IBM PC i PC DOS

Tomasz Kozdrowicz

IBM PC
i PC DOS

Tomasz Kozdrowicz

IBM PC i PC DOS

Wydawnictwa Komunikacji i Łączności
Warszawa 1988

Opiniodawca: dr inż. *Jan Bielecki*
Okładkę projektowała: *Cecylia Staniszewska*
Redaktor: mgr inż. *Lech Pilarowski*
Redaktor techniczny: *Maria Dylewska*

681.3

Książka zawiera opis podstawowych konfiguracji komputerów zgodnych z IBM PC/XT/AT, szczegółowy opis poleceń systemu operacyjnego PC DOS, jak również opis wybranego oprogramowania użytkowego (tzw. nakładki systemowe - Xtree, Norton Commander, Norton Utilities, SideKick).
Książka jest podstawowym kompendium wiedzy dla zarówno początkujących, jak i zaawansowanych użytkowników komputerów zgodnych z IBM PC.

ISBN 83-206-0843-0

© Copyright by
Wydawnictwa Komunikacji i Łączności
Warszawa 1988

Wydawnictwa Komunikacji i Łączności
Warszawa 1988
Wydanie I. Nakład 49 650+350 egz.
Ark. wyd. 10. Ark. druk. 13/17,3 A/1
Papier offset. kl. III 70 g, rola 70 cm
Oddano do wykonania diapozytywów w lipcu 1988 r.
Podpisano do druku we wrześniu 1988 r.
Zam. P/67/88. Zam. druk. 1366/88 U-54/192
Olsztyńskie Zakłady Graficzne im. S. Pieniężnego
10-417 Olsztyn, ul. Towarowa 2

SPIS TREŚCI

Wprowadzenie 7

1. Budowa, konfiguracja i zastosowanie 11

2. PC DOS - wprowadzenie do systemu 33

3. Sam na sam z DOSem 49
3.1 Uruchomienie systemu 50
3.2 Kopiowanie dyskietek i zbiorów 52
3.3 Uruchomienie systemu z twardym dyskiem 61
3.4 Podstawowe czynności 64
3.5 Praca z katalogami 67
3.6 Edycja poleceń DOSu - klawisze edycyjne i sterujące 76
3.7 Potoki i filtry w DOSie 81
3.8 Polecenia DOSu 83
3.9 Konfiguracja systemu 117
3.10 Przetwarzanie wsadowe - zbiory typu BAT 126
3.11 Skrócony opis edytora EDLIN 133
3.12 Skrócony opis edytora pakietu SideKick 139

4. Efektywne wykorzystanie systemu operacyjnego 146
4.1 O co chodzi 146
4.2 Podstawy 146
4.3 Co dalej 148
4.4 Przykłady, czyli sztuczki i chwyty 149

5. Oprogramowanie 181

5.1 Xtree 181
5.2 Norton Commander 190
5.3 Norton Utilities - Advanced Edition 198

Dodatek A - zestaw znaków ASCII 204

Dodatek B - zestaw znaków graficznych IBM 205

Dodatek C - zestaw kodów sterujących Epson ESC/P ... 207

WPROWADZENIE

Tytuł książki jest właściwie skróconym spisem treści. Na początku zapoznamy się ogólnie z budową komputerów zgodnych (kompatybilnych) z IBM PC XT/AT, a potem omówimy działanie i sposób korzystania z systemu operacyjnego PC DOS. Na zakończenie zajmiemy się oprogramowaniem, które pracuje pod nadzorem systemu PC DOS. W zasadzie większa część książki jest przeznaczona dla początkujących użytkowników komputerów, ale również osoby bardziej doświadczone powinny znaleźć dla siebie coś nowego i ciekawego (zwłaszcza w rozdziale 4).

Komputery osobiste zgodne z IBM PC są urządzeniami przeznaczonymi dla indywidualnego użytkownika. Rysunek 1 przedstawia typowy wygląd takiego komputera w standardowej konfiguracji sprzętowej wraz z opisem jego podstawowych, widocznych na zewnątrz elementów.

Ponieważ w Polsce sprzęt komputerowy nie jest tani, indywidualny użytkownik, dla którego jest on przeznaczony, musi przestrzegać określonych zasad postępowania. Ma to na celu uzyskanie długiej żywotności sprzętu i uchronienie go przed awariami, będącymi wynikiem nieprawidłowej obsługi.

Troska o komputer jest uzasadnioną troską użytkownika o samego siebie. Żaden szef na pewno nie pochwali swojej sekretarki za uszkodzenie karty CGA, spowodowane niewłaściwym włączaniem komputera, czy za uszkodzenie twardego dysku spowodowane potrąceniem maszyny. Dlatego należy pamiętać o pewnych zaleceniach, dotyczących eksploatacji komputerów osobistych.

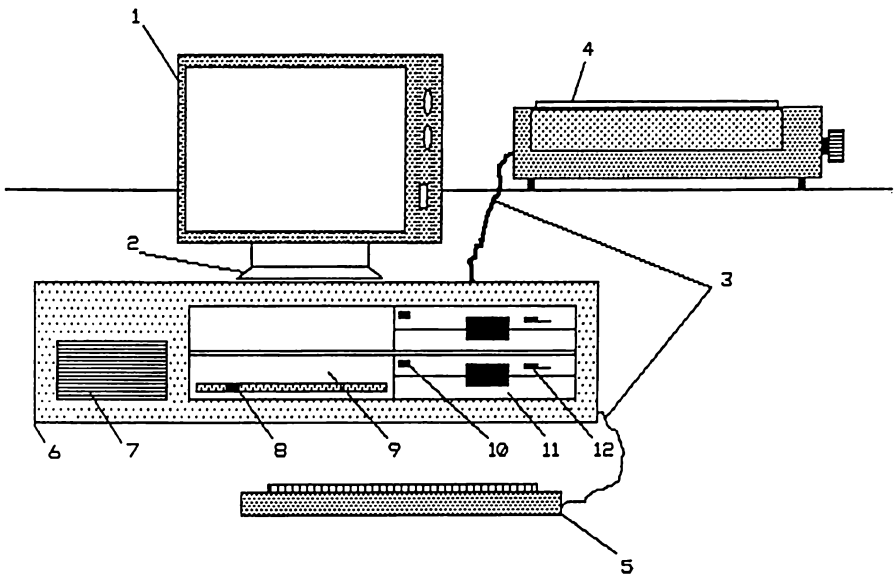
Na początku nie wszystko, co zostało poniżej napisane, musi wydawać się jasne (dla lepszego zrozumienia opisu zasad eksploatacji należy dokładnie zapoznać się z rys. 1). Nie trzeba się tym jednak przejmować - chodzi głównie o to, aby niedoświadczony użytkownik nie zepsuł sprzętu przed jego poznaniem. Wyjaśnienie nie dla wszystkich zrozumiałych terminów zostało zawarte w rozdziale 1. Przejdźmy zatem do opisu podstawowych zaleceń dotyczących eksploatacji komputerów.

● Najpierw włączamy zasilanie jednostki centralnej, a dopiero kilka sekund później - monitor. Niewskazane jest włączenie monitora na stałe i uruchamianie go razem z całym systemem za pomocą włącznika sieciowego jednostki centralnej. Taki zabieg jest jednoznaczny z włączaniem i wyłączaniem monitora przez wkładanie i wyjmowanie z gniazda sieciowego przewodu zasilającego. Przez włączenie monitora razem z jednostką centralną można uszkodzić np. kartę CGA, która - jak wiadomo ze smutnych doświadczeń wielu osób - jest wrażliwa na taki sposób postępowania. Naturalnie system wyłączamy w kolejności odwrotnej - najpierw monitor, a dopiero po nim jednostkę centralną.

● Należy pamiętać, że przed wyłączeniem systemu dobrze jest (nawet tak na wszelki wypadek) zaparkować twardy dysk. Może to wydawać się nadmiarem ostrożności, ale jednak lepszy jest chyba nadmiar ostrożności w obsłudze urządzenia za kilkaset tysięcy złotych, niż jej brak. Uszkodzenie twardego dysku może narazić nas nie tylko na konieczność jego wymiany, ale i np. na rekonstruowanie siatki płac lub stanu magazynu.

● Mimo, że twardy dysk jest zamknięty hermetycznie, przestrzegałbym przed paleniem papierosów w pomieszczeniu z komputerem i w ogóle przed eksploatacją komputera w pomieszczeniach zadymionych, zapylonych i wilgotnych. Takie warunki pracy znacznie skracają żywotność całego sprzętu i powodują zwiększenie jego awaryjności.

● Nie można zapominać o czyszczeniu głowic w napędach miękkich dysków. Trudno jest sprecyzować jak często należy czyścić głowice - jest to uzależnione od intensywności korzystania z napędów i rodzaju stosowanych dyskietek (firmowe, czy też "no name" w białych kopertach), ale zalecałbym czyszczenie co dwa, trzy miesiące. Najlepiej jest zlecić czyszczenie



Rys.1. Wygląd ogólny komputera w standardowej konfiguracji
1 - monitor, 2 - podstawa monitora, 3 - przewody łączące drukarkę i klawiaturę z jednostką centralną, 4 - drukarka, 5 - klawiatura, 6 - jednostka centralna, 7 - otwory wentylacyjne i głośnik, 8 - dioda sygnalizująca pracę twardego dysku, 9 - twardy dysk (płyta czołowa), 10 - dioda sygnalizująca pracę stacji miękkich dysków, 11 - stacja (napęd) miękkich dysków, 12 - drzwi stacji miękkich dysków

głowic serwisowi firmy, od której komputer został kupiony, ponieważ operacja taka jest związana z wyjęciem napędu z obudowy komputera. Najbardziej skuteczne jest przemycie głowic czystym spirytusem - zestawy czyszczące oferowane przez różne firmy (Verbatim, 3M, PerfectData, Inmac i inne) nie cieszą się najlepszą opinią wśród fachowców tak rodzimych, jak i zagranicznych (vide miesięcznik PC Resource, wrzesień 1987).

- Klawiaturę należy chronić przed kurzem. Trzeba ją przykryć kawałkiem plastiku, jeżeli nie jest używana. To samo dotyczy monitora - po pracy można go przykryć od góry (tam, gdzie są otwory wentylacyjne) również jakimś kawałkiem folii plastikowej.

● W czasie transportu komputera należy nie tylko zaparkować twardy dysk, ale i zabezpieczyć głowice napędów dysków miękkich przez włożenie do nich płaskich kawałków kartonu (dostarczanych razem z komputerem) i zamknięcie drzwi napędu. Chroni to głowice przed nadmiernymi wibracjami i uderzaniem o siebie (w napędach dwustronnych, obecnie najpopularniejszych).

● Dyskietek nie należy wyginać, wkładać "na siłę" do napędu, kłaść na urządzeniach emitujących fale elektromagnetyczne (monitory, telewizory, kolumny głośnikowe itp.), ani wystawiać na działanie zbyt wysokich lub zbyt niskich temperatur. Tworzywo sztuczne (mylar), z którego są wykonane dyskietki, jest czułe na zmiany temperatury, co może spowodować uszkodzenie warstwy nośnika magnetycznego.

Większość z powyższych wskazówek wydaje się być oczywista. Należy jednak pamiętać, że dobrze jest nie tylko znać te zalecenia, ale i stosować je podczas codziennej eksploatacji komputera. Na pewno nie jest to zbyt męczące, a może uchronić każdego z nas przed przykrymi niespodziankami w postaci awarii sprzętu lub straty wielu godzin pracy i wielu kilobajtów danych.

1. BUDOWA, KONFIGURACJA I ZASTOSOWANIE

Komputery osobiste typu IBM PC (czyli modele PC Junior, XT, XT286, AT) mają budowę modułową. Oznacza to, że komputer taki składa się ze szkieletu, do którego można dołączać różne urządzenia i moduły, nazywane powszechnie kartami. Tym szkieletem jest płyta główna (ang. motherboard), nazywana również płytą-matką, umieszczana przeważnie poziomo na spodzie obudowy komputera. Znajdują się na niej przede wszystkim:

- procesor centralny komputera,
- układy wspomagające, jak sterowniki przerwań, sterowniki bezpośredniego dostępu do pamięci (ang. direct memory access, w skrócie DMA), układy obsługi urządzeń zewnętrznych (peryferyjnych), gniazdo do podłączenia klawiatury,
- układy pamięci stałej (ang. read only memory, w skrócie ROM), czyli pamięci, z której można tylko odczytać wcześniej zapisaną informację. Układy te zawierają w sobie oprogramowanie tzw. BIOSu (skrót od ang. basic input/output system, czyli podstawowy system wejścia i wyjścia) i niekiedy interpreter języka BASIC,
- pamięć operacyjna o dostępie swobodnym (ang. random access memory, w skrócie RAM),
- miejsce na koprocesor arytmetyczny (procesor wspomagający procesor główny komputera przy szczególnie pracochłonnych zadaniach numerycznych),
- gniazda rozszerzające (ang. expansion slots), czyli tzw. "grzebienie", w które wkłada się "szyny" kart zmieniających lub uzupełniających konfigurację komputera.

Procesor centralny jest właściwie sercem komputera. Przyjmuje on informacje dostarczane przez otaczające go urządzenia wejścia i odsyła je do urządzeń wyjścia po przetworzeniu. Szybkość pracy i sprawność przetwarzania informacji przez komputer w dużej mierze zależą od zastosowanego procesora centralnego. W komputerach IBM PC, XT i AT oraz w tzw. klonach (w miarę wiernych kopiach produktów firmy IBM) najczęściej są stosowane poniżej opisane procesory.

- Intel 8088. Jest to procesor zastosowany w pierwszych modelach IBM PC i XT. Jest on procesorem 16-bitowym z 8-bitową szyną danych co oznacza, że informacja we "wnętrzu" procesora jest przetwarzana w porcjach po 16 bitów, a przyjmowana i wysyłana w porcjach po 8 bitów. Nie jest to zatem procesor w pełni 16-bitowy, czego wynikiem jest niezbyt szybka praca (wąskie gardło stanowi 8-bitowe przesyłanie informacji). Zwiększenie częstotliwości pracy tego procesora ze standardowych 4.77 MHz do 8 MHz powoduje zwiększenie szybkości działania o 60%. Im większa częstotliwość pracy procesora, tym więcej działań wykonuje on w jednostce czasu. Niestety, wraz z prędkością pracy procesora rośnie jego cena. Jest to szczególnie widoczne w przypadku procesorów wyższej klasy (Intel 80286, Intel 80386) i koprocesorów arytmetycznych.
- Intel 8086. Jest to właściwie Intel 8088 z 16-bitową szyną danych. Pracując z częstotliwością 8 MHz, zwiększa on trzykrotnie szybkość wykonywania operacji w porównaniu z procesorem 8088 przy 4.77 MHz. Interesująca jest oferta firmy NEC, która produkuje procesory V20 i V30 (odpowiedniki Intel 8088 i 8086) pracujące z częstotliwościami 4.77, 8 i 10 MHz. Ponieważ procesory te mają nieco usprawnioną architekturę, pracują one ok. 15% ÷ 20% szybciej, niż ich Intelowskie odpowiedniki. Najbardziej chyba godny uwagi jest procesor V30 10MHz, który w klasie XT umożliwia osiągnięcie szybkości działania komputera porównywalnej z szybkościami niektórych komputerów klasy AT (ponad pięć razy szybciej, niż komputer z procesorem Intel 8088-4.77 MHz).
- Intel 80286. Jest również procesorem w pełni 16-bitowym, ale ma większe możliwości programowania niż Intel 8086. Podstawowe jego zalety to sprzętowa, czyli "wbudowana" w

procesor zdolność do pracy wielozadaniowej oraz możliwość użycia do 1 GB (gigabajta) pamięci wirtualnej (pamięci "widzianej" przez oprogramowanie - programowi może "wydawać się", że ma do dyspozycji 1 GB pamięci, mimo że komputer ma fizycznie dołączoną mniejszą pamięć). Dzięki ulepszeniom sprzętowym i przeważnie dużej częstotliwości pracy (do 16 MHz) procesora 80286, komputery, w których jest on stosowany, mogą pracować nawet 15-krotnie szybciej niż komputery z procesorem Intel 8088. Należy jednak przestrzec przed stosowaniem bardzo dużych częstotliwości pracy jednostki centralnej, gdyż już przy 12 MHz mogą powstać problemy w komunikacji pomiędzy procesorem a otaczającymi go urządzeniami zewnętrznymi. Wiele kart rozszerzających (modułów zwiększających możliwości funkcjonalne komputera) może pracować z częstotliwościami nie przekraczającymi 8 MHz (jest to częstotliwość stosowana w oryginalnych IBM AT). Jeżeli karta taka zostanie zainstalowana w maszynie 12 MHz, to nie zawsze zdąży ona skomunikować się z procesorem komputera i może albo nie zadziałać, albo spowodować awarię systemu. Dlatego często jest stosowane zmniejszanie częstotliwości pracy komputera w momencie odwoływania się do kart rozszerzających. Duża częstotliwość pracy wymaga też stosowania szybkich pamięci (z czasem dostępu 120, a nawet 100 nanosekund). Dzięki temu procesor nie musi czekać przy pobieraniu czy zapisywaniu informacji do pamięci - operacje te są wykonywane w jednym cyklu pracy procesora (ang. zero-wait state memory, czyli pamięć bez stanu oczekiwania procesora). Zastosowanie wolniejszych (a tym samym tańszych) pamięci nie pozwala na pełne wykorzystanie dużej częstotliwości pracy procesora. Trzeba zauważyć, że ceny szybkich pamięci są bardzo wysokie, dlatego ich zastosowanie jest ograniczone. Przeważnie procesory pracujące z dużymi częstotliwościami podczas komunikowania się z pamięcią przechodzą w stan oczekiwania (ang. one-wait state memory) do momentu pobrania z pamięci potrzebnej informacji. Takie rozwiązanie zmniejsza szybkość pracy komputera, ale jednocześnie pozwala na zastosowanie znacznie tańszych układów pamięci.

- Intel 80386. Jest to procesor w pełni 32-bitowy pracujący z

częstotliwością 16, 20 lub 24 MHz, stosowany w drogich maszynach najwyższej klasy, o dużej mocy przetwarzania. Obecnie istniejące oprogramowanie (tak systemowe, jak i użytkowe) nie wykorzystuje nawet niewielkiej części możliwości procesora 80386. W chwili obecnej komputery osobiste z tym procesorem są najszybszymi i najdroższymi maszynami w swojej klasie, przy czym pełne wykorzystanie ich możliwości nastąpi dopiero po wprowadzeniu na rynek wielozadaniowych systemów operacyjnych (np. długo już oczekiwanego OS/2 firmy Microsoft) i odpowiedniego oprogramowania.

Podsumowując nasze informacje o procesorach można powiedzieć, że osoba kupująca komputer powinna dobrze zdawać sobie sprawę z tego, do jakich zadań sprzęt będzie stosowany i wybrać taką maszynę, która spełni określone wymagania za rozsądną cenę. Nie zawsze warto płacić dużo więcej za dużo szybszy komputer, jeżeli nie da to użytkownikowi rzeczywiście znaczących korzyści. Poza tym nie w każdym zastosowaniu komputera szybkość pracy jego procesora centralnego jest najważniejsza. Niekiedy decyduje szybkość urządzeń zewnętrznych, czasami możliwość rozszerzenia konfiguracji, a najczęściej po prostu księgowy, który z pewnością potrafi wytłumaczyć każdemu niedowiarkowi, że sprzęt tani jest znacznie lepszy od żadnego.

Sterownik przerwań znajdujący się na płycie głównej, ma za zadanie przyjmować sygnały przerwań od różnych urządzeń komputera. Urządzenia te przez wysłanie sygnału przerwania żądają pozwolenia na skomunikowanie się z procesorem centralnym, zajętym w tym czasie np. wykonywaniem obliczeń dla użytkownika. Sterownik przerwań porządkuje sygnały przerwań według określonej hierarchii i umożliwia kolejnym urządzeniom skomunikowanie się z procesorem centralnym przez przerywanie co jakiś czas jego pracy. Komputery klasy XT mają sterowniki pozwalające na obsługę do 8 przerwań, natomiast AT - do 15. Z tego względu komputery klasy AT mają większe możliwości instalowania urządzeń wymagających korzystania z przerwań.

Sterowniki DMA (ang. direct memory access, czyli bezpośredni dostęp do pamięci) umożliwiają określonym urządzeniom bezpośredni dostęp do pamięci komputera i korzystanie z niej bez obciążania procesora centralnego.

Układ obsługi urządzeń zewnętrznych spełnia wiele zadań - od sprawdzania konfiguracji ustawienia mikroprzełączników na płycie głównej do przyjmowania sygnałów wejściowych z klawiatury.

Ponieważ ani sterowniki przerwań, ani sterowniki DMA, ani układy obsługi urządzeń zewnętrznych nie są elementami decydująco wpływającymi na relację ceny i szybkości komputera, czy też na możliwości zwiększania funkcjonalności sprzętu, nie będziemy się nimi już dłużej zajmować.

Dodatkowa pamięć ROM - o ile (oprócz BIOSu) w ogóle jest instalowana na płycie głównej - zawiera w sobie przeważnie interpreter BASICu. Nie jest to rzecz konieczna i na dobrą sprawę raczej nieważna przy wyborze komputera. BIOS jest programem obsługującym komunikację wszystkich urządzeń wejścia/wyjścia komputera z procesorem centralnym oraz ładującym system operacyjny po włączeniu maszyny. Program ten jest zapisany w pamięci ROM lub EPROM (ang. erasable programmable ROM, czyli programowalna pamięć ROM, z możliwością zmiany zapisanej w niej informacji). Oryginalny BIOS firmy IBM został opatentowany, w związku z czym wszystkie kopie komputerów tego producenta są zaopatrywane w układy innych firm (ze względu na brak możliwości wykorzystania BIOSu oryginalnego). Jeżeli działanie BIOSu jakiegoś innego producenta nie jest całkowicie zgodne z działaniem BIOSu oryginalnego, mogą wystąpić kłopoty ze zgodnością sprzętu i dlatego nie wszystkie programy napisane dla IBM będą pracować poprawnie. Z tego powodu przy wyborze komputera należy zwrócić uwagę na to, aby jego BIOS był programem znanym i sprawdzonym (najbardziej godne zaufania są produkty firmy Phoenix).

Pamięć operacyjna RAM jest wykorzystywana w czasie pracy przez procesor centralny do przechowywania programu oraz informacji przetworzonej i czekającej na przetworzenie. Obecnie niezbyt szybkie układy pamięci RAM nie wpływają zasadniczo na cenę komputera i dlatego można sobie pozwolić na kupno maszyny z zainstalowaną pamięcią co najmniej 512 KB. Wiele programów potrafi bardzo efektywnie wykorzystać taką pamięć, a trzeba także wspomnieć, że niektóre programy po prostu wymagają dużej pamięci (Windows Graph 1.0 wymaga 512 KB, DESQview

- 640 KB, InHouse Accountant - 384 KB itd.). Ważnym parametrem jest czas dostępu do pamięci. czyli czas niezbędny do pobrania z komórki pamięci jej zawartości. W niektórych zastosowaniach komputerów czas dostępu do pamięci i liczba stanów oczekiwania decydująco wpływają na szybkość wykonywania pewnych programów, w innych nie są czynnikami znaczącymi - zagadnienie współpracy między układami pamięci a procesorem centralnym zostało omówione nieco szerzej w tej części niniejszego rozdziału, która dotyczy procesora Intel 80286.

Koprocesor arytmetyczny jest układem współpracującym z procesorem centralnym i zwiększającym jego moc przy obliczeniach zmiennoprzecinkowych. Koprocesory używane w komputerach osobistych IBM i z nimi zgodnych to różne odmiany 80-bitowego Intela 8087 i 80287. Jak łatwo zgadnąć, Intel 8087 współpracuje z 8088 lub 8086, a 80287 - z 80286. W momencie pisania niniejszej książki dopiero jest zapowiadane wejście na rynek koprocesora Intel 80387 (przewidywana cena - $1300 za Intel 80387-20 MHz), który będzie wspomagać 32-bitowy 80386.

Koprocesor arytmetyczny nie jest tanim elementem systemu komputerowego i przed jego kupnem trzeba dobrze zastanowić się, czy rzeczywiście będzie on intensywnie wykorzystywany. Należy zauważyć, że w większości przypadków cena koprocesora nie jest wliczona w cenę całego sytemu. Dlatego zakup koprocesora musi być dobrze przemyślany. Na pewno układ ten powinien posiadać każdy użytkownik, który wykonuje dużo obliczeń numerycznych (przede wszystkim prace inżynierskie, żmudne obliczenia w fizyce itp.). Korzyść wynikająca z zastosowania koprocesora jest dwojaka. Po pierwsze, uzyskuje się znacznie krótszy (5 do 10 razy) czas obliczeń, przy czym obliczenia te są dokładniejsze od obliczeń dokonanych bez koprocesora (jakie to ma znaczenie dla inżyniera nie trzeba chyba wyjaśniać). Po drugie, część oprogramowania dostępna na polskim rynku, nie działa bez koprocesora (dotyczy to szczególnie kompilatorów języka Fortran oraz niektórych języków symulacyjnych). Co prawda popularne Turbo C emuluje działanie koprocesora, a Turbo Pascal (wersja 4.0) potrafi obejść się bez niego, ale odbywa się to znowu kosztem czasu obliczeń. Sprawa koprocesora nie jest zatem tak prosta, jak na pierwszy rzut oka mogłoby wydawać się. Intel 8087 daje niepodważalne korzyści w kom-

puterach typu XT, ale zastosowanie 80287 w AT niekiedy może być dyskusyjne. Ponieważ w AT (w odróżnieniu od XT) koprocesor jest taktowany impulsami z niezależnego generatora, przeważnie nie pracuje on tak szybko, jak jednostka centralna. Z tego powodu przyspieszenie pracy komputera nie jest tak duże, jak w XT i wydanie kilkuset tysięcy złotych na koprocesor dający znikome korzyści staje się sprawą problematyczną. Zatem przed zakupem AT z koprocesorem najrozsądniej jest przetestować szybkość pracy maszyny z i bez 80287 i na tej podstawie określić rzeczywistą efektywność danego rozwiązania sprzętowego. Należy także pamiętać, że na rynku występuje po kilka odmian (generacji) 8087 i 80287. Jeżeli chodzi o 8087, to w grę wchodzą przede wszystkim układy generacji HMOS3, tzn. C8087 (pracujący z częstotliwościami do 5 MHz), C8087-2 (do 8 MHz) oraz C8087-1 (do 10 MHz). Przy zakupie trzeba pamiętać, że częstotliwość, z jaką pracuje nasz XT, w żadnym przypadku nie może być większa od znamionowej częstotliwości koprocesora. Dołączenie C8087 do systemu pracującego z częstotliwością 8 MHz spowodowałoby grzanie się koprocesora i jego nieprawidłową pracę (przekłamania). Z powodu znacznej wrażliwości na podwyższoną temperaturę pracy nikogo nie namawiam do zakupu koprocesorów poprzedniej generacji, tj. HMOS1 (C8087-4, C8087 -6, C8087-3 oraz C80287-3).

W maszynach typu AT generator taktujący koprocesor pracuje z reguły z częstotliwościami 4, 6, 8 i 10 MHz (w najszybszych wersjach). Koprocesor przeważnie pracuje z częstotliwością o 2 MHz mniejszą, niż jednostka centralna. I tak, Intel C80287-3 może pracować z częstotliwością do 5.33 MHz (zaleca się 4 MHz - generacja HMOS1!), C80287-6 - do 6 MHz oraz C80287-8 - do 8 MHz. Przed zainstalowaniem C80287-6 w komputerze pracującym z częstotliwością 8 MHz należy koniecznie sprawdzić częstotliwość taktowania koprocesora, a następnie upewnić się, czy rzeczywiście przyspieszenie pracy, wynikające z zastosowania koprocesora, jest warte wydawanych pieniędzy.

Ostatnim elementem płyty głównej, który pozostał nam do omówienia, są tzw. *gniazda rozszerzające* (ang. expansion slots) nazywane również otworami rozszerzającymi lub - w gwarze profesjonalistów - slotami. Gniazda te pozwalają na podłączenie do komputera w bardzo prosty sposób dodatkowych urządzeń.

Istnieją różne typy gniazd rozszerzających: 8-bitowe (gniazdo typu XT), 16-bitowe (gniazdo typu AT) i 32-bitowe w komputerach z procesorem Intel 80386. Z reguły komputery XT są wyposażane w 8 gniazd 8-bitowych, natomiast AT - w 2 do 4 gniazd 8-bitowych i 4 do 6 gniazd 16-bitowych. Na naszym rynku sporadycznie można spotkać maszyny z 12 gniazdami. Jeżeli chodzi o znaczenie liczby gniazd dla użytkownika, to zasada oceny jest prosta - im więcej gniazd, tym bardziej można komputer rozbudować. Rozbudowa polega na wkładaniu dodatkowych kart (modułów) w gniazda. Karty są czymś w rodzaju klocków, z których - przy wykorzystaniu płyty głównej jako swego rodzaju fundamentu - można zbudować dokładnie taki komputer, jaki jest nam potrzebny. Możliwości rozbudowy są bardzo duże. Można instalować karty sterowników różnych wyświetlaczy, karty obsługujące różne urządzenia dodatkowo dołączane do komputera, od prawie 3 lat można kupić twardy dysk na karcie (o pojemności nawet do 80 MB), istnieje możliwość zainstalowania karty z lepszym procesorem, który zastąpi lub wspomoże procesor centralny z płyty głównej. Dostępne są karty rozszerzeń pamięci, karty umożliwiające zainstalowanie komputera w sieci czy dołączenie do niego modemu. Można zaryzykować stwierdzenie, że w praktyce rzadko kiedy w pełni wykorzystuje się możliwości rozbudowy sprzętu. Odnosi się to szczególnie do warunków polskich, w których komputer w pełnej konfiguracji kosztuje fortunę, a w dodatku instalowanie niektórych urządzeń zewnętrznych nie ma większego sensu - np. modemów, czyli urządzeń sprzęgających komputery przez sieć telefoniczną. Polska sieć telefoniczna wnosi tak duże zakłócenia do przesyłanej nią informacji, że stosowanie modemów jest wielce problematyczne.

Dla użytkownika jednym z najważniejszych elementów komputera jest jego *monitor*. Monitor musi być obsługiwany przez *sterownik znakowy* lub *znakowo-graficzny*. Karta sterownika jest umieszczana w jednym z gniazd rozszerzających. W chwili obecnej w Polsce można spotkać się z kilkoma rodzajami kart graficznych. Kartą najpopularniejszą jest tzw. Hercules. Konkuruje z nią znacznie droższa i bardziej atrakcyjna EGA (w różnych odmianach), natomiast na straconej pozycji jest CGA. Karty MDA właściwie już nie spotyka się. Zanim wyjaśnimy so-

bie, co oznaczają te dla niektórych czytelników tajemnicze skróty, najpierw kilka słów o wyświetlaczach w ogóle. Wyświetlacz (ang. video display unit, w skrócie VDU) składa się ze sterownika (zwanego kartą graficzną lub znakową albo kontrolerem graficznym) oraz monitora. Sterownik monitora jest interfejsem (czyli - używając polskiej terminologii - łączem lub sprzęgiem) między procesorem centralnym komputera a jego monitorem. Pozwala na wyświetlenie na monitorze informacji przesyłanych przez procesor centralny do standardowego urządzenia wyjścia. Treść informacji wyświetlanej na monitorze nie zależy od sterownika. Sterownik determinuje jedynie sposób, w jaki dana informacja może być przedstawiona. Określa on kształt liter, możliwości użycia kolorów, rozdzielczość grafiki itd. Łatwo dostępnym na naszym rynku sterownikiem monitora jest tzw. EGA (ang. Enhanced Graphics Adapter), łącząca w sobie zalety CGA (ang. Color Graphics Adapter) i Herculesa (ang. Hercules Graphics Card, w skrócie HGC), czyli kolorową grafikę i dużą rozdzielczość. Sterowniki typu MDA (ang. Monochrome Display Adapter) pacują tylko w trybie znakowym i nie dają żadnych możliwości graficznych, co powoduje całkowity brak zainteresowania tymi sterownikami. Sprawą oczywistą jest, że karta określonego sterownika graficznego musi współpracować z odpowiednim monitorem, umożliwiającym właściwe wykorzystanie możliwości tej karty. Wyświetlacz w nowo kupowanym sprzęcie należy dobierać bardzo starannie. Jeżeli, skuszeni kolorami i dobrze opracowanymi programami demonstracyjnymi, zdecydujemy się na wydanie dużej ilości pieniędzy, to może okazać się, że nasza wspaniała EGA jest wykorzystywana do pracy z bazami danych i edytorami tekstów, a jej możliwości graficzne są najzupełniej zbędne. Zbytnia oszczędność grozi tym, że żaden projektant nie będzie chciał korzystać z pomocy komputera przy robieniu rysunków technicznych, ze względu na za małą rozdzielczość grafiki i związaną z tym obawę pogorszenia wzroku.

Przy wyborze wyświetlacza należy więc zwrócić uwagę na jego przydatność, cenę i parametry techniczne. Decydujacymi parametrami są:

- matryca znaku,
- możliwości graficzne sterownika,

- częstotliwość odchylania (poziomego i pionowego) monitora,
- średnica plamek luminoforu w kineskopie monitora,
- możliwość emulacji trybu pracy innych wyświetlaczy.

Matryca znaku określa ilość punktów, z których jest tworzona pojedyncza litera na monitorze. Karta CGA ma matrycę znaku 8x8, Hercules - 9x14, EGA - 8x14. Trzeba powiedzieć, że tekst z CGA nie jest na tyle czytelny, aby karta ta umożliwiała poważną pracę z tekstami. Jeżeli zatem nie potrzebujemy kolorowej grafiki, optymalnym rozwiązaniem jest Hercules. Daje on bardzo wyraźny i doskonale czytelny tekst oraz spore możliwości w grafice monochromatycznej (jednokolorowej). Jest to jednocześnie rozwiązanie o wiele tańsze od karty EGA, która oprócz wyraźnego tekstu i dobrej grafiki, daje nam możliwość korzystania z kolorów (tak w trybie tekstowym, jak i w graficznym), co przy dużej rozdzielczości powoduje konieczność współpracy z drogim, wysokiej klasy monitorem kolorowym.

Możliwości graficzne sterownika są określone przez rozdzielczość i ilość kolorów, które mamy do dyspozycji i które możemy jednocześnie wyświetlić na ekranie. Rozdzielczość mówi o liczbie punktów wyświetlanych na monitorze w poziomie i w pionie. CGA daje 640x200 punktów przy 2 kolorach lub 320x200 punktów przy 4 kolorach wybranych z palety 16. Hercules ma rozdzielczość 720x348 punktów (brak kolorów), a standardowa EGA - 640x350 przy 4 kolorach na ekranie. Niektórzy niezależni producenci wytwarzają karty zgodne z kartami EGA firmy IBM, zwiększając ilość pamięci RAM zainstalowanej na karcie. 128 KB RAM umożliwia wykorzystanie pełnej rozdzielczości przy 16 kolorach, natomiast 256 KB RAM dodatkowo pozwala na płynne przesuwanie (ang. scrolling) obrazu.

Częstotliwość odchylania poziomego monitora to częstotliwość, z jaką strumień elektronów przemiata ekran. Determinuje to stabilność i ostrość obrazu. Im częstotliwość ta jest większa, tym stabilniejszy i ostrzejszy jest obraz. Z reguły do kart CGA stosuje się monitory o częstotliwości odchylania poziomego równej 15.75 kHz, do Herculesa - 18.432 kHz; częstotliwość odchylania pionowego dla obu typów kart standardowo wynosi 50 ÷ 60 Hz. Spotyka się karty EGA, które wymagają stosowania monitorów akceptujących częstotliwość odchylania po-

ziomego dochodzącą nawet do 35 kHz. Takimi monitorami są np. NEC MultiSync oraz nieco lepszy Princeton Ultrasync. NEC MultiSync ustanowił pewien standard wśród monitorów wysokiej klasy. Był on pierwszym monitorem na rynku, który pozwalał na współpracę ze sterownikami o różnych częstotliwościach odchylania - zakup tego urządzenia dawał pewność, że nawet przy przewidywanej zmianie karty sterownika wyświetlacza (np. z CGA na EGA) razem z nią nie trzeba będzie zmieniać monitora.

Średnica plamek luminoforu wpływa na wyrazistość i ziarnistość obrazu. W monitorach monochromatycznych pojedyncze plamki luminoforu mają średnicę z reguły 0.36 mm ÷ 0.39 mm, monitory do kart typu EGA nie powinny mieć więcej niż 0.31 mm (spotyka się konstrukcje osiągające 0.21 mm - np. Wyse Technology). Im mniejsza średnica plamki luminoforu, tym wyraźniejszy i mniej ziarnisty obraz.

Możliwość emulacji trybu pracy innych wyświetlaczy polega na tym, że sterownik danego wyświetlacza z punktu widzenia oprogramowania zachowuje się tak, jak sterownik innego wyświetlacza. Powinna obowiązywać tu zasada "zgodności w dół". Oznacza to, że Hercules może emulować CGA, EGA może emulować CGA i Herculesa (oryginalna EGA firmy IBM emuluje CGA i MDA), karty PGC (o których za chwilę) niezależnych producentów powinny dawać możliwość emulacji trybu pracy tak Herculesa, jak CGA i EGA. Ta "zgodność w dół" jest bardzo ważna ze względu na istniejące oprogramowanie. W zaraniu dziejów komputerów osobistych IBM w grafice królowała karta CGA, dlatego większość starszych programów wykorzystujących grafikę została dostosowana do parametrów właśnie karty CGA. Jeżeli ktoś ma kartę EGA lub Hercules, to program dostosowany do CGA albo nie da się uruchomić, albo nie będzie działał prawidłowo. Dlatego dla takiego programu nasza EGA lub (częściej) Hercules musi zachowywać się tak, jak CGA. Istnieje również oprogramowanie wykorzystujące możliwości graficzne Herculesa. Jeżeli mamy CGA, to raczej nic się nie da zrobić i program po prostu nie będzie działał. W przypadku dobrej karty EGA niezależnego producenta emulacja trybu pracy HGC na pewno będzie możliwa.

Należy zauważyć, że emulacja trybu pracy innego sterownika graficznego może być realizowana programowo lub sprzętowo. Z

reguły emulacja CGA na Herculesie jest dokonywana programowo (po uruchomieniu programu emulującego Hercules dla innego oprogramowania zachowuje się jak CGA), natomiast karta EGA jest przełączana za pomocą odpowiedniego oprogramowania użytkowego (nie emulacyjnego) w tryb pracy odpowiadający CGA lub HGC. Takie przełączenie można również wykonać za pomocą mikroprzełączników znajdujących się na karcie sterownika.

Karty CGA, Hercules i EGA są w chwili obecnej najbardziej w Polsce rozpowszechnionymi kartami sterowników wyświetlaczy. Oprócz nich występują również rozwiązania bardziej specjalistyczne, jak np. PGC (ang. Professional Graphics Controler) firmy IBM czy też Hercules InColor i Hercules Plus firmy Hercules Computer Technology. PGC pozwala nam korzystać z 256 kolorów (oferując 4096 barw do wyboru) przy grafice o rozdzielczości 640x480 punktów. Hercules InColor imponuje rozdzielczością 720x348 przy 16 kolorach wybranych z palety 64 oraz możliwością zdefiniowania w pamięci operacyjnej sterownika własnych 3072 znaków w 16 kolorach (lub 12288 znaków w 2 kolorach).

Niektóre sterowniki wyświetlaczy dają możliwość zobrazowania na ekranie zamiast standardowych 80 kolumn w 25 wierszach, 132 kolumny w 44 wierszach (można też spotkać wyświetlacze 90 kolumn na 38 wierszy - np. Hercules Graphics Card Plus). Teoretycznie ma to umożliwić umieszczenie większej ilości informacji tekstowej na ekranie, szczególnie przy pracy z arkuszami kalkulacyjnymi (Multiplan) i pakietami zintegrowanymi (Lotus 1-2-3, Symphony). Wydaje mi się jednak, że nawet mimo stosowania największej rozdzielczości operacja taka nie ma większego sensu. Informacji, owszem, jest rzeczywiście znacznie więcej na ekranie, ale jest to informacja słabo czytelna i ogromnie "zagęszczona". Znalezienie i odczytanie czegokolwiek staje się trudne, ekran przypomina gazetę zadrukowaną bardzo drobną czcionką. Dlatego rozsądniejsze jest chyba pozostanie przy rozwiązaniach bardziej klasycznych i tańszych. Trzeba bowiem pamiętać, że możliwość wyświetlenia 132 kolumn tekstu w 44 wierszach przeważnie jest reklamowana jako coś, bez czego w ogóle nie można się obejść i za co warto zapłacić. Pamiętajmy więc, żeby nie dać się zwariować i przed kupnem sprawdzić, czy to, co kupujemy, nadaje się do użytku.

Opisy ciekawych rozwiązań sterowników wyświetlaczy można mnożyć w nieskończoność, ale pamiętając o polskich realiach trzeba stwierdzić, że nie ma to większego sensu, ponieważ rodzi tylko niepotrzebne frustracje. Po prostu im większa rozdzielczość i im więcej kolorów tym wyższa cena. Karta Hercules InColor w USA (stan na wrzesień 1987) kosztuje około $180, odpowiedni monitor około $500, razem czyni to $680. Do tego dochodzi transport, zarobek osoby sprzedającej pośrednikowi, prowizja pośrednika - w sumie prawie 3 miliony złotych. Hercules monochromatyczny kosztuje $40, razem z monitorem $120. Dlatego do przetwarzania tekstów, baz danych i do celów ściśle obliczeniowych trzeba stosować monochromatyczną kartę Hercules. Przy stosunkowo niskiej cenie mamy bardzo wyraźny tekst i spore możliwości graficzne, jesteśmy jedynie pozbawieni kolorów. Stosowanie kart EGA, PGC czy Hercules InColor ma sens w przypadku prowadzenia wyrafinowanych prac projektowych i intensywnym używaniu pakietów CAD (ang. Computer Aided Design, czyli projektowanie wspomagane komputerem).

Jednym ze standardowych urządzeń wyjścia jest (obok monitora) *drukarka*. Drukarki do komputerów osobistych IBM są podłączane przeważnie przez łącze równoległe znajdujące się albo na karcie sterownika wyświetlacza, albo na karcie wielofunkcyjnej (ang. multi i/o, czyli karta z wieloma wejściami/wyjściami) przy użyciu kabla wykorzystującego wtyki standardu Canon i Centronics. Łącze równoległe drukarki jest również nazywane równoległym portem drukarki (ang. parallel printer port).

Polski rynek drukarek w znacznej części został opanowany przez drukarki igłowe firmy Star. Stało się tak dzięki obecności drukarek Stara w ofercie wszystkich wysyłkowych firm tajwańskich oraz dzięki ich niskim cenom, nawet w Europie Zachodniej. Mówiąc o niskich cenach mam na myśli niskie ceny w porównaniu z cenami drukarek innych firm (IBM, Epson, Panasonic, Alps). Sporadycznie występują inne drukarki - można spotkać sprzęt Amstrad/Schneider, Okidata, QMS, Seikosha. Ze względu na bardzo wysoką jakość najbardziej cenione są drukarki Alps (w pełni profesjonalne, przystosowane do pracy non-stop) i Epson. Drukarki laserowe to Hewlett-Packard, Toshiba, C.Itoh, Taxan itd. Inne urządzenia drukujące to drukarki

atramentowe, termiczne i elektrofotograficzne (laserowe drukujące w kolorach).

Zasada działania drukarek igłowych jest bardzo prosta. Drukowanie następuje przez uderzanie szeregu igieł (młoteczków) drukujących w nasączoną tuszem taśmę. Taśma w momencie uderzenia pozostawia na papierze ślad. Igły znajdują się w głowicy drukującej, która przesuwa się wzdłuż papieru. Po wydrukowaniu jednej linii następuje zmiana wiersza i ponowny druk. W niemal wszystkich drukarkach (oprócz np. polskich D-100) druk odbywa się w obie strony, tzn. podczas przesuwu głowicy z lewej strony na prawą, jak również - z prawej na lewą. Głowicą drukarki steruje komputer przez wyjście, do którego drukarka została podłączona.

Podstawowymi parametrami drukarki są:
- szybkość druku,
- jakość druku,
- możliwości graficzne,
- głośność pracy przy drukowaniu,
- szerokość wałka drukarki (ilość znaków w wierszu),
- trwałość głowicy drukującej i taśmy barwiącej.

Szybkość drukowania producenci podają w znakach na sekundę (ang. characters per second, w skrócie cps) dla dwóch rodzajów druku: zwykłego (ang. draft) i podwyższonej jakości, nazywanego również drukiem korespondencyjnym (ang. near letter quality, w skrócie NLQ). Sposób jednoznacznego testowania prędkości drukowania nie został jeszcze ustalony (nie ma odpowiednich norm), dlatego też dane podawane przez producentów należy traktować jako przybliżone. Ostatnio zostały podjęte pewne działania, zmierzające do wprowadzenia miarodajnych testów pracy drukarek (test EPPT, skrót od ang. European Printer Performance Test). Przeciętnie drukarki 9-igłowe pracują z prędkością 120 ÷ 200 cps (w trybie korespondencyjnym 30 ÷ 60 cps). Drukarki 24-igłowe osiągają prędkość do 360 cps, a np. drukarka atramentowa SQ-2500 firmy EPSON - 540 cps.
Prędkość pracy drukarek laserowych podaje się w stronach A4 na minutę - przeważnie jest to 6 ÷ 10 stron na minutę (dla porównania z drukarkami igłowymi dodajmy, że strona znormalizowanego maszynopisu zawiera 1800 znaków). Oczywiście jakość

wydruku z drukarki laserowej jest znacznie lepsza niż z tradycyjnej drukarki igłowej, ale większe są również koszty zakupu i eksploatacji.

Jakość wydruku jest zależna od matrycy znaku - w drukarkach 9-igłowych matryca ma wymiary przeważnie 9x9 punktów (w trybie korespondencyjnym od 16x20 do 18x24 punkty) Drukarki 24-igłowe mają matrycę o wymiarach rzędu 9x23 punkty (w trybie korespondencyjnym 29x23 punkty). Ważna jest wizualna ocena efektów pracy drukarki (szczególnie przy ocenianiu drukarek o podobnych parametrach technicznych) i niekiedy ona decyduje o wyborze sprzętu.

Możliwości graficzne drukarek określa ich rozdzielczość (rozumiana tak samo, jak w przypadku monitorów), podawana w punktach na cal (ang. dots per inch, w skrócie dpi). Dla drukarek igłowych największa obecnie osiągana rozdzielczość to 360 x180 dpi, a dla drukarek laserowych standardem jest 300x300 dpi. Drukarki igłowe w trybie graficznym pracują nieznośnie wolno, dlatego jeżeli urządzenie ma być służyć głównie zastosowaniom graficznym, konieczne staje się kupno plotera. Jeżeli drukarka laserowa ma pracować w trybie graficznym na całej stronie A4, to konieczne staje się zainstalowanie w niej ponad 1 MB pamięci operacyjnej, co jest bardzo kosztowne. W skład standardowego wyposażenia drukarek laserowych wchodzi przeważnie 512 KB pamięci RAM, co pozwala na korzystanie z grafiki na połowie formatu A4. W niektórych zastosowaniach nie jest to wystarczające.

Głośność pracy drukarki jest szczególnie ważna, jeżeli w jednym pomieszczeniu ma jednocześnie pracować kilka urządzeń. Poziom głośności jest najrzadziej chyba podawanym parametrem drukarek igłowych. Dzieje się tak dlatego, że drukarki igłowe, niezależnie od wszystkiego, pracują głośno, a np. laserowe i atramentowe cicho. Przeciętny poziom hałasu wytwarzany przez drukarkę igłową wynosi około 57 dB, drukarki atramentowe nie przekraczają 45 dB, a laserowe - 50 dB.

Szerokość wałka drukarki ma znaczenie przede wszystkim w zastosowaniach graficznych i wymagających wyprowadzania na papier dużych ilości danych (tekstu, liczb, rysunków). Myślę tutaj o korzystaniu z programów takich, jak AutoCad czy Mul-

tiplan. Drukarki igłowe są produkowane z wałkami o dwóch szerokościach - 10" (25 cm) i 15" (40 cm). To samo dotyczy drukarek atramentowych, natomiast posiadacze drukarek laserowych przeważnie są skazani na jeden rozmiar papieru - A4. Jest oczywiste, że przy tej samej rozdzielczości drukarka o szerszym wałku może wydrukować znacznie większy rysunek i znacznie więcej tekstu, niż drukarka o wałku węższym. Niekiedy można zrezygnować z plotera na rzecz drukarki z wałkiem 40 cm, jeżeli nie zależy nam na posługiwaniu się kolorami. Trzeba też pamiętać o tym, że każda zmiana linii podczas drukowania oznacza zmniejszenie średniej prędkości drukowania, tak więc drukarki z szerokimi wałkami, przy wykorzystaniu całej szerokości papieru, są nieco szybsze od drukarek z wąskimi wałkami.

Trwałość głowicy drukującej w drukarkach igłowych wynosi 100 ÷ 250 mln. znaków (przy pracy w trybie normalnym), taśma barwiąca wystarcza na wydrukowanie średnio 2.5 mln znaków. Parametry drukarek atramentowych są znacznie lepsze niż igłowych - przeciętna trwałość głowicy drukującej (dysz) to 2000 mln. znaków, a jeden pojemnik z atramentem wystarcza na wydrukowanie około 6 mln. znaków. Poza tym jakość wydruku z drukarki atramentowej jest bardzo dobra do ostatniej chwili, natomiast drukarka igłowa drukuje coraz mniej wyraźnie już po kilku tygodniach eksploatacji. Nadmierna oszczędność przy zaopatrywaniu się w nowe taśmy barwiące prowadzi do powstawania nieczytelnych wydruków. Trwałość głowic drukarek laserowych jest określana w liczbie stron A4 i przeciętnie wynosi 200000 ÷ 300000 stron. Co prawda wystarcza to nawet na kilka lat eksploatacji (przy drukowaniu 3÷4 tys. stron miesięcznie), ale wymiana głowicy kosztuje tyle, co cała, bardzo dobra drukarka igłowa. Poza tym po wydrukowaniu każdych 3÷4 tys. stron należy uzupełnić toner (proszek będący czynnikiem barwiącym w drukarkach laserowych i kserografach), który do najtańszych nie należy.

Podsumowując rozważania o drukarkach, można powiedzieć, że drukarki laserowe w warunkach polskich (na razie - miejmy nadzieję) nie mają racji bytu (przede wszystkim ze względu na bardzo duże koszty zakupu i eksploatacji). Drukarki atramentowe są niewątpliwie rozwiązaniem technicznie lepszym od

igłowych. Istnieje jednak pewne "ale". Drukarka atramentowa najlepiej pracuje na papierze dobrym gatunkowo, ze znalezieniem którego w odpowiedniej ilości na pewno każdy miałby kłopoty. Nie można zapominać również o tym, że co jakiś czas trzeba kupić nowy pojemnik z atramentem. Niestety do tej pory nasze rodzime firmy, zajmujące się sprzedażą i serwisem drukarek, właściwie w ogóle nie zauważają drukarek atramentowych, coraz popularniejszych w wielu innych krajach. Z tego też powodu zakup pojemników z tuszem, czy jakakolwiek awaria i związana z nią konieczność naprawy, mogą przysporzyć wielu kłopotów. Dlatego rozwiązaniem najpewniejszym wciąż jeszcze wydaje się być zakup drukarki igłowej, może niezbyt wyrafinowanej technicznie i na pewno zbyt głośnej, ale ze sprawnym (no, nie przesadzajmy - istniejącym) serwisem i łatwo dostępnymi taśmami barwiącymi.

Często zamiast drukarki lepiej jest zastosować *ploter* (ang. plotter), czyli urządzenie kreślące. Ploter podłączany jest przeważnie do złącza komunikacji szeregowej (RS 232C), rzadziej wykorzystywane są złącza równoległe (Centronics 8-bitowy). Rysunek jest tworzony na podłożu (może to być papier, przezroczysta folia do rzutników itp.) za pomocą pisaka umieszczonego w chwytaku ramienia plotera. Pisak może być automatycznie zmieniony przez ploter na inny podczas rysowania - dzięki temu łatwo realizuje się zmianę koloru na rysunku. Podstawowa różnica między drukarką a ploterem jest taka, że drukarka tworzy obraz na papierze metodą rastrową (punkt po punkcie), natomiast ploter wykorzystuje metodę wektorową (linia po linii). Z tego względu plotery doskonale nadają się do tworzenia np. rysunków technicznych i w ogóle obrazów, które nie zawierają dużej ilości zabarwionych płaszczyzn.
Głównym zastosowaniem ploterów stało się ich wykorzystanie do wyprowadzania wyników pracy programów wspomagających projektowanie (CAD). Ponieważ w technice jest wymagana bardzo duża precyzja, konstrukcje ploterów umożliwiają wykonywanie dokładnych rysunków (z rozdzielczością co najmniej 0.05 mm) na dużych powierzchniach. Co prawda spotyka się plotery, które nie akceptują arkuszy papieru (czy też folii) o formacie większym niż A4 (np. Epson HI-80), ale do poważnych zastosowań nadają się tylko urządzenia pozwalające na kreślenie rysunków o for-

macie A0. Prędkość kreślenia przyzwoitego plotera nie powinna być mniejsza, niż 125 mm/s (5 ips, czyli 5 cali na sekundę). Ważna jest też możliwość korzystania z kolorów - przeważnie do dyspozycji mamy 4- lub 16-kolorowy zestaw pisaków. Wykorzystanie wszystkich możliwości, dawanych przez konstrukcję dobrego plotera, jest uzależnione od tego, czy używane przez nas oprogramowanie potrafi odpowiednio sterować ploterem. Dlatego trzeba kupować taki ploter, który potrafi emulować działanie poularnego modelu urządzenia znanej firmy (np. ploter 7475 Hewlett-Packard). Na największe zaufanie zasługują produkty takich firm, jak właśnie Hewlett-Packard, Roland, United Innovations i Houston Instrument/Ametek.

Skaner (ang. scanner) nie jest produktem popularnym na polskim rynku. Wiele osób, które mogłyby wykorzystać skaner w pracy, nie zna nawet możliwości takiego urządzenia. Dlatego też postaram się nieco dokładniej omówić działanie i zastosowanie skanerów. Ogólnie rzecz biorąc, skaner pozwala na przekształcenie obrazu istniejącego na papierze w postać elektryczną. Wyróżniane są trzy podstawowe typy skanerów: skanery graficzne, optyczne czytniki znaków (ang. optical character reader, w skrócie OCR) i skanery transmisyjne.

Skanery graficzne służą do przetwarzanie obrazów (elementy znakowe występujące na obrazach są traktowane również jako grafika) na grafikę bitową (raster lub mapa bitów), która może być wykorzystana i przetworzona przez oprogramowanie komputera. Istnieje oprogramowanie "tłumaczące" grafikę rastrową na grafikę wektorową. Takie przetworzenie grafiki jest konieczne wtedy, kiedy obraz wczytany za pomocą skanera chcemy trochę poprawić, pokolorować i narysować ploterem. I tak np. skaner graficzny może przenieść fotografię lotniczą jakiegoś terenu do pamięci komputera, co umożliwi późniejszą obróbkę tego obrazu za pomocą odpowiedniego oprogramowania, a potem narysowanie kolorowej mapy na ploterze. Obraz przenoszony przez skaner do pamięci komputera jest przeważnie czarno-biały (każdy punkt obrazu z reguły jest reprezentowany przez 4 bity, co daje $2^4 = 16$ stopni szarości). Skanery graficzne, umożliwiające przenoszenie także kolorów, należą jeszcze do rzadkości.

Skanery typu OCR umożliwiają przenoszenie tekstów z papieru do komputera. Tekst odczytany przez OCR jest zapamiętywany na dysku w postaci zbioru tekstowego. Przeważnie istnieje możliwość wybrania formatu tego zbioru (np. zgodnego z formatem, w jakim zbiory tekstowe zapisują takie programy, jak WordStar czy dBase III+). Do niedawna dla skanerów OCR czytelnych było tylko kilka krojów czcionek maszyn do pisania. W chwili obecnej możliwe jest czytanie tekstu wydrukowanego na dobrej drukarce igłowej, nie sprawia też kłopotu tekst uzyskany za pomocą fotoskładu. Większość OCR oferuje możliwość rozpoznawania kodu prążkowego. Niestety nie spotkałem się jeszcze z tego typu urządzeniem, które rozpoznawałoby polskie znaki, chociaż jest to właściwie kwestia tylko odpowiedniego oprogramowania. Jeżeli na rynku pojawią się takie rozwiązania, to mogą one wielu osobom zaoszczędzić mozolnego wprowadzania danych (przepisywanie tekstów ze sprawozdań, zestawień, artykułów), poprawek (prace wydawnicze) itd.

Skanery transmisyjne służą do przenoszenia obrazów drogą telekomunikacyjną. Pierwsze konstrukcje tego typu pojawiły się w Japonii już dobrych kilka lat temu. Zainteresowanie skanerami transmisyjnymi, przesyłającymi (często kolorową) grafikę liniami telefonicznymi lub teleksowymi, wyniknęło w Japonii stąd, że teksty japońskie można transmitować jedynie w postaci graficznej. Ze względu na mnogość znaków nie dają one zakodować się w tak prosty sposób, jak teksty angielskie (cały zestaw znaków ASCII składa się z 256 elementów, z czego na litery i cyfry przypada mniej niż 100).

Dla polskiego użytkownika najbardziej interesujące są niewątpliwie skanery graficzne. Ponieważ nie ma miejsca, żeby w pełni przeanalizować parametry techniczne skanerów graficznych i opisać wpływ tych parametrów na wybór urządzenia, zainteresowanych czytelników odsyłam do miesięcznika PC Magazine, Vol 6, Nr 17, z 13 października 1987 roku, gdzie zagadnieniom skanerów zostało poświęconych ponad 100 stron. Są tam również omówione podstawowe zagadnienia konstruowania algorytmów przetwarzania grafiki rastrowej na wektorową. Na zakończenie tych kilku uwag o skanerach chciałbym jeszcze polecić kilka typów tych atrakcyjnych urządzeń, a mianowicie Datacopy

JetReader, Datacopy 730, Hewlett-Packard Scan Jet, IBM 3118 Scanner Model 20 i Canon IX-12 Image Scanner.

Twarde dyski są obecnie najatrakcyjniejszym łatwo dostępnym rodzajem zewnętrznej pamięci masowej. Umożliwiają one gromadzenie olbrzymiej ilości danych, oferując jednocześnie szybki do nich dostęp. Praktycznie nie spotyka się już w dzisiejszych czasach twardych dysków o pojemnościach mniejszych niż 10 MB. Taki dysk kosztuje w USA ok. $140, a razem ze sterownikiem jego cena nie przekracza $220. W Polsce żaden użytkownik komputera nie interesuje się sprzętem oferującym mniej niż 20 MB. Trzeba jednak pamiętać, że 10 MB wystarcza na przechowywanie prawie 6000 stron znormalizowanego maszynopisu A4. Zatem jak na sprzęt, który ma służyć jednej osobie w przedsiębiorstwie, jest to chyba wystarczająco dużo. Pewną nieprawidłowością, dającą się zauważyć w instytucjach (a także na wyższych uczelniach) posiadających więcej niż jeden komputer, jest dążenie do umieszczenia kopii wszystkich posiadanych programów w każdym komputerze. Dochodzi do sytuacji, w której na twardym dysku mamy dziesiątki świetnych programów, a żadnego z nich nie można szerzej wykorzystać, ponieważ dysk jest zapełniony "po brzegi" i nie ma na nim miejsca na zapisanie rezultatów jakiejkolwiek poważniejszej pracy. Najrozsądniejszym wyjściem z sytuacji jest zapisanie na każdym z komputerów różnych programów i takie zorganizowanie pracy, aby osoby korzystające z tego samego oprogramowania wzajemnie sobie nie przeszkadzały. Tego typu rozwiązanie pozwoliłoby na uzyskanie sporych oszczędności - mimo ciągle na świecie spadających cen twardych dysków, na zakup takiego urządzenia w Polsce trzeba przeznaczyć 1.2 ÷ 1.6 mln. zł. (dysk 20 MB wraz ze sterownikiem). Twardy dysk jest jednym z głównych elementów całego systemu komputerowego, a jego najważniejszymi cechami powinny być niezawodność i szybkość pracy. Niezawodność zapewnia nabycie dysku dobrego producenta, jak np. Plus Development, NEC, Seagate, Core, Bernoulli. Szybkość pracy twardego dysku jest określona przez średni czas dostępu (ang. average access time) do zapisanych danych. Dla komputerów klasy XT standardem jest obecnie 65 ms, dla AT - 40 ms. Najlepsze rozwiązania w klasie AT osiągają obecnie 17 ms. Szybkość pracy twardego dysku ma decydujące znaczenie

wtedy, gdy mamy zamiar korzystać z oprogramowania intensywnie z nim współpracującego. Jeżeli natomiast twardy dysk będzie przeznaczony przede wszystkim do przechowywania danych i oprogramowania (komputer będzie wykorzystywany np. do intensywnych prac obliczniowych i operacja czytania/pisania na dysk będzie odbywać się dwa razy dziennie), to nie ma najmniejszego sensu płacić za zmniejszenie czasu dostępu o kilka milisekund. Nie można bowiem zapominać, że cena twardego dysku jest wykładniczo proporcjonalna nie tylko do jego pojemności, ale także do jego szybkości. Pracę twardego dysku nadzoruje sterownik (zbudowany niemal zawsze z układów firmy Western Digital). Sterownik taki jest elementem płyty głównej komputera typu AT, natomiast w komputerach typu XT i baby-AT (AT ze zmniejszoną płytą główną) trzeba instalować oddzielną kartę sterownika. Przeważnie sterownik twardego dysku jest umieszczany na karcie razem ze sterownikami miękkich dysków lub na karcie wielofunkcyjnej. Zatem przy zakupie sprzętu należy zwrócić uwagę na to, w jakim komputerze dysk będzie używany oraz czy i jaki sterownik (na jakiej karcie) należy dokupić. Ostatnio pojawiło się na rynku nieco dysków ze sterownikami typu RLL, które pozwalają na lepsze wykorzystanie pojemności dysku. Na twardym dysku, dającym z konwencjonalnym sterownikiem 20 MB, można uzyskać 30 MB przy użyciu sterownika typu RLL. Trzeba jednak przyznać, że napęd dysku pracuje wtedy na granicy swoich możliwości i z tego powodu rozwiązanie napęd konwencjonalny/sterownik RLL nie cieszy się opinią zupełnie niezawodnego.

Ostatnim typem urządzenia peryferyjnego, który omówimy, będą myszy (ang. mouse) i digitizery (ang. digitizer). Należą one do grupy urządzeń wejścia i są określane mianem urządzeń wskazujących (ang. pointing devices).
Zasada działania myszy polega na przekazywaniu ruchu urządzenia (przemieszczanego na płaszczyźnie przez użytkownika) kursorowi, znajdującemu się na ekranie monitora komputera. Odpowiednie oprogramowanie interpretuje położenie kursora i stan klawiszy (są przeważnie dwa lub trzy) na myszce.
Myszy, ze względu na zasadę translacji swojego ruchu i położenia na sygnały elektryczne, dzielą się na mechaniczne (w

tym kołowe), optomechaniczne i optyczne. Ponieważ myszy mechaniczne należą do najbardziej zawodnych, a optyczne potrzebują do pracy specjalnej płytki z rastrem, która zajmuje na biurku cenne miejsce i w ogóle nie jest urządzeniem zbyt wygodnym, najbardziej atrakcyjne są myszy optomechaniczne.
Digitizer to coś w rodzaju dokładnej myszy, której pozycja jest bardzo precyzyjnie określana - dlatego digitizery służą do współpracy z programami typu CAD. Za ich pomocą można w miarę łatwo przenieść współrzędne punktów z rysunku na papierze do rysunku tworzonego na komputerze.
Digitizer jest niezbędny przy korzystaniu z oprogramowania wspomagającego prace projektowe, natomiast zwykła mysz przydaje się, jeżeli jest wykorzystywane oprogramowanie sterowane za pomocą okien i zestawów menu. Również graficznie zorientowane pakiety oprogramowania (np. Desktop Publishing, w skrócie DTP) pozwalają na efektywne wykorzystanie myszy. Trzeba jednak przyznać, że jeżeli określony program współpracuje równie dobrze z kilkoma rodzajami myszy, to wybór myszy staje się kwestią osobistych upodobań użytkownika - po prostu jedna osoba woli mysz o kształcie prostopadłościennym, a innej wygodniej pracuje się z myszą o bardziej giętkich liniach. Tak myszy, jak i digitizery mogą albo być podłączane do złącza komunikacji szeregowej, albo wymagać włożenia karty swojego sterownika do jednego z gniazd rozszerzających. Urządzenia dwóch czołowych producentów myszy (Microsoft i Mouse Systems) oferowane są tak w konfiguracji "serial" (podłączenie do RS 232C), jaki i "bus" (wykorzystanie jednego z gniazd rozszerzających). Każde oprogramowanie, które dopuszcza wykorzystanie myszy, będzie z pewnością współpracować z myszami obydwu tych firm. Największym konkurentem dla Microsoftu i Mouse Systems jest obecnie firma Logitech. Oferuje ona doskonałe, optomechaniczne myszy, emulujące działanie produktów Microsoftu i Mouse Systems i dostarczane z bardzo bogatym oprogramowaniem. To oprogramowanie decyduje o tym, że zakup myszy Logitechu jest najlepszym rozwiązaniem.

2. PC DOS - WPROWADZENIE DO SYSTEMU

System operacyjny PC DOS jest zestawem programów umożliwiających zarządzanie komputerem i informacją zawartą w pamięci zewnętrznej i operacyjnej maszyny. Skrót DOS pochodzi od angielskiej nazwy Disk Operating System (Dyskowy System Operacyjny). Oprócz PC DOS spotykamy również MS DOS. MS DOS jest systemem operacyjnym stworzonym przez firmę Microsoft (stąd przedrostek MS w nazwie systemu). System ten sprzedawany przez IBM z komputerami tej firmy został nazwany PC DOS (PC jest skrótem od ang. Personal Computer - nazwy pierwszego komputera osobistego IBM).

Historia MS DOS i PC DOS zaczyna się w roku 1981, kiedy to na rynek wprowadzono IBM PC z PC DOS 1.0. W kilka miesięcy później wprowadzono MS DOS 1.25 i PC DOS 1.1, dające możliwość używania dyskietek o podwójnej gęstości zapisu (napędy dwustronne). Wraz z wejściem na rynek IBM XT (marzec 1983) pojawiają się kolejne wersje - MS DOS 2.0 i PC DOS 2.0. Najpoważniejszą zmianą w stosunku do wersji poprzednich była możliwość obsługi twardego dysku i tworzenia katalogów o hierarchicznej strukturze. Kolejne wersje MS DOS (2.01, 2.11, 2.25) oraz PC DOS (2.1) pozwoliły na stosowanie narodowych zestawów liter (tzw. zestaw Latin2, zawierający także polskie litery, nigdy nie został zrealizowany przez IBM). Usunięto z nich także pewne błędy, które pojawiły się w wersjach poprzednich. Rok 1984 przyniósł MS DOS 3.0 i PC DOS 3.0, wprowadzone na rynek wraz z komputerami IBM AT. Te wersje systemów operacyj-

nych dawały możliwość korzystania ze stacji dyskietek 1.2 MB z twardych dysków o dużych pojemnościach oraz pozwalały zaadresować większą pamięć. Kolejna wersja 3.1 pozwalała na pracę w sieciach Microsoft, a wersja 3.2 na korzystanie z nowych nośników informacji (dyskietki 3.5 cala, dyski optyczne). Dodano również kilka nowych poleceń systemowych, np. **XCOPY**. Najmłodsze dziecko Microsoftu - MS DOS 3.30 - jest systemem znacznie bardziej od swoich poprzedników rozbudowanym. Pozwala na szybszy i wygodniejszy dostęp do podkatalogów (nowe polecenie **APPEND**), obsługuje stacje dysków 3.5 cala 1.44 MB, daje możliwość podziału twardego dysku na kilka dysków logicznych (dzielenie na partycje dysków o dużych pojemnościach nie jest już konieczne). Opis zmian w stosunku do wersji 3.2 jest dość obszerny, jego szczegółowe przytoczenie nie jest w tej chwili celowe. Ostatnio jest zapowiadany system operacyjny OS/2 (Operating System 2), który ma pracować na najnowszej generacji komputerów osobistych IBM PS/2 (Personal System/2) i komputerach zgodnych z IBM, opartych na procesorach Intel 80286 i Intel 80386, oraz umożliwiać wykonywanie do 16 programów jednocześnie (wielozadaniowość).

W chwili obecnej w Polsce najbardziej rozpowszechniony jest niewątpliwie DOS 3.10, również dosyć często można spotkać się z DOSem 3.20. Dlatego w rozdziale trzecim, omawiającym dokładniej system operacyjny, skoncentrujemy się na DOSie 3.10, a także przyjrzymy się bliżej kilku atrakcyjnym rozszerzeniom DOSu 3.20. System operacyjny dla przeciętnego użytkownika jest właściwie integralną częścią komputera. Stanowi on osobowość maszyny zgłaszającej się znakiem gotowości i potulnie czekającej na polecenia. Trzeba jednak zdać sobie sprawę, że od momentu włączenia sprzętu do pojawienia się zgłoszenia systemu dzieje się wiele rzeczy. Żeby zrozumieć procesy zachodzące podczas "ładowania się" (wczytywania) systemu do pamięci, należy najpierw poznać strukturę tego systemu i otoczenia jednostki centralnej (procesora centralnego) komputera. W publikacjach firmy Microsoft można znaleźć podział systemu operacyjnego MS DOS na następujące części:

- BIOS,
- system właściwy,
- procesor poleceń.

BIOS, jak już wcześniej zostało powiedziane, to podstawowy system obsługi wejść i wyjść (Basic Input Output System). Jest to program zapisany w pamięci ROM (lub EPROM) i dostarczany razem z komputerem przez producenta. Zawiera on w sobie procedury obsługi standardowych urządzeń wejścia i wyjścia, jak np. monitora i klawiatury (czyli konsoli CON), stacji dysków, drukarki (PRN), łącza komunikacji szeregowej. Procedury te są wywoływane przez oprogramowanie systemu właściwego, który dzięki temu nie musi komunikować się bezpośrednio ze sterownikami poszczególnych urządzeń zewnętrznych (sterowniki te mogą być różne w różnych komputerach). System zleca poszczególne operacje wejścia/wyjścia programowi BIOSu, który jest dostosowany do porozumiewania się na poziomie sprzętowym z dołączonymi urządzeniami peryferyjnymi. Użytkownik może również zainstalować (podłączyć do systemu komputerowego) urządzenia, których obsługa nie jest przewidziana przez oprogramowanie zawarte w BIOSie. Taka instalacja może być realizowana np. programowo za pomocą polecenia DEVICE w zbiorze CONFIG.SYS, wykorzystywanym podczas procesu inicjalizacji systemu przez zbiór IBMBIO.COM (wykorzystanie programowych sterowników urządzeń) - ale o tym później. Standardowe urządzenia wejścia/wyjścia DOSu to:

CON: (od ang. console) jako urządzenie wejścioa oznacza klawiaturę.

CON: jako urządzenie wyjścia oznacza monitor.

A: oznacza pierwszy napęd dysków miękkich.

B: oznacza drugi napęd dysków miękkich.

C: oznacza napęd twardego dysku.

D: oznacza z reguły RAM-dysk (urządzenie symulowane w pamięci operacyjnej komputera), zwany również dyskiem wirtualnym, lub drugi twardy dysk.

COM1: (od ang. communication) to pierwsze łącze (port) komunikacji szeregowej, czyli urządzenie we/wy przesyłające informację szeregowo, zwane również AUX: (od ang. auxillary).

COM2: to drugie łącze komunikacji szeregowej.

LPT1: (lub PRN:) oznacza pierwsze łącze równoległe (przesyłające strumienie informacji jednocześnie, czyli równolegle), będące wyjściem na drukarkę.

LPT2: i LPT3: to kolejne wyjścia równoległe na drukarki.

NUL: oznacza urządzenie testujące, które nie istnieje w rzeczywistości. Służy do sprawdzania możliwości przesłania informacji. Jeżeli symuluje urządzenie wejścia, to natychmiast po uruchomieniu generuje znacznik końca przesyłu informacji. Jeżeli symuluje urządzenie wyjścia, to pozwala na wykonanie operacji pisania danych bez rzeczywistego ich przesyłania.

Dwukropek występujący w nazwach urządzeń we/wy można pomijać przy odwoływaniu się do nich. Nie jest to dozwolone w przypadku odwoływania się do urządzeń typu blokowego (o swobodnym dostępie do informacji), czyli przede wszystkim do nazw stacji dysków.

Ważną częścią BIOSu są procedury POST i procedury ładowania. POST oznacza Power On Self Test (jest to procedura testująca procesor centralny, pamięć RAM, sterowniki monitora, klawiatury, stacji dysków miękkich itd.). Procedury ładowania umożliwiają wprowadzenie do pamięci z dyskietki tej części systemu operacyjnego, która czuwa nad dalszym procesem jego wczytywania.

Jądrem systemu operacyjnego są dwa zbiory programowe: IBMBIO.COM i IBMDOS.COM (w oryginalnym systemie Microsoftu są to IO.SYS i MSDOS.SYS). Na dyskietce zbiory te występują z atrybutami "system" i "hidden" (systemowe i ukryte).

IBMBIO.COM to program częściowo przejmujący i rozszerzający funkcje BIOSu (np. jeżeli chodzi o sterowanie urządzeniami zewnętrznymi zainstalowanymi przez użytkownika) po wczytaniu do pamięci operacyjnej. Procedury IBMBIO.COM są właściwie procedurami niskiego poziomu (poziomu sprzętowego) - tak samo, jak procedury BIOSu.

IBMDOS.COM jest już programem wyższego rzędu, który korzysta z pośrednictwa BIOSu i IBMBIO.COM podczas komunikacji z urządzeniami zewnętrznymi. Procedury, które można wywołać z IBMDOS.COM są pod względem funkcjonalnym dość złożone (w przeciwieństwie do elementarnych instrukcji IBMBIO.COM i BIOSu) i służą bezpośrednio takim programom DOSu, jak np. **FORMAT** czy **DISKCOPY**, oraz programom użytkowym. Z tego względu IBMDOS.COM może być określony mianem łącza programowego między procedu-

rami niskiego poziomu a bardziej kompleksowymi poleceniami wysokiego poziomu.

Procesor poleceń COMMAND.COM jest programem interpretującym rozkazy otrzymane z klawiatury. To, co wprowadzamy do komputera za pomocą klawiatury, jest pobierane przez COMMAND.COM i porównywane najpierw z listą poleceń wewnętrzych (czyli takich, które są zawarte w pamięci operacyjnej komputera po wczytaniu systemu operacyjnego), a potem jest wyszukiwany program (w katalogach zdefiniowanych w poleceniu PATH) o nazwie zgodnej z nazwą polecenia wprowadzonego z klawiatury. COMMAND.COM w każdym z wyszczególnionych katalogów szuka programów o podanej nazwie i rozszerzeniach (ostatnie trzy znaki pełnej nazwy programu występujące po kropce) kolejno COM, EXE oraz BAT. Jeżeli program taki nie zostanie znaleziony, system wyświetli komunikat dobrze znany każdemu:

Bad command or file name

czyli

Niewłaściwe polecenie lub nazwa zbioru

Jeżeli program taki zostanie znaleziony, to system przystąpi do wczytania i wykonania go.

Po tym krótkim wprowadzeniu dotyczącym budowy systemu operacyjnego PC DOS i MS DOS możemy spróbować poznać procesy zachodzące między włączeniem komputera do sieci a pojawieniem się zgłoszenia systemu. Jak już wcześniej zostało powiedziane, jedną z funkcji BIOSu jest wykonanie procedury POST (samoczynnego testu) - czyli zbadanie poprawności działania najważniejszych układów maszyny i ich inicjalizacja, oraz przetestowanie konfiguracji sprzętu. Konfiguracja sprzętowa w komputerach typu XT jest zakodowana w ustawieniu mikroprzełączników na płycie głównej, natomiast w komputerach typu AT informacje dotyczące konfiguracji są zapisane w pamięci RAM (podtrzymywanej bateryjnie). Informacje te wprowadzane są tam przez wyspecjalizowane programy diagnostyczne. Pewna część dołączanych dodatkowo do podstawowej konfiguracji komputera urządzeń po włączeniu sprzętu dokonuje autoinicjalizacji i

zgłasza swoją obecność systemowi. Jeżeli w BIOSie jest zawarte rzeczywiście dobre oprogramowanie, to potrafi ono wykryć nawet nieprawidłowe ustawienie mikroprzełączników, opisujące konfigurację inną od istniejącej. Po wykonaniu procedury POST sterowanie pracą komputera zostaje przekazane programowi ładującemu (zapisanemu oczywiście również w BIOSie). Program ten szuka pierwszego sektora (ang. boot sector) najpierw na dyskietce w stacji A:, a potem na twardym dysku i wczytuje z niego dyskowy program ładujący (disk bootstrap). Dyskowy program ładujący przejmuje nadzór nad systemem, po czym sprawdza, czy na dysku znajduje się kopia DOSu. Czytelnikowi może wydawać się dziwne, czemu jeden program ładujący wczytuje inny program ładujący i jemu z kolei przekazuje nadzór nad dalszym uruchamianiem systemu. Dzieje się tak, ponieważ istnieją różne systemy operacyjne, używające między innymi różnych formatów (sposobów zorganizowania zapisu informacji) dyskietek. Program ładujący wczytany z odpowiedniej dyskietki systemowej "wie", jak zorganizowane jest formatowanie i w jaki sposób czytać informacje z dyskietek. Zatem praca różnych systemów musi zaczynać się tak samo - od uruchomienia przez BIOS programu ładującego tegoż systemu. Dzięki temu każdy, kto dobrze zna funkcje BIOSu i działanie sterowników poszczególnych urządzeń peryferyjnych (a także jest bardzo odważny) może zrealizować swój własny system operacyjny.

Po tej krótkiej dygresji możemy znowu zająć się naszym przed chwilą włączonym komputerem. Dyskowy program ładujący sprawdza obecność kopii DOSu na dyskietce (lub twardym dysku) przez przeczytanie głównego katalogu i odnalezienie tam zbiorów IBMBIO.COM i IBMDOS.COM w takiej właśnie kolejności (powyższy opis dotyczy PC DOS, w MS DOS szukane są IO.SYS i MSDOS.SYS). Jeżeli tych zbiorów nie ma, to na ekranie monitora pojawia się (również wszystkim bardzo dobrze znany) komunikat:

```
Non-System disk or disk error
Replace and strike any key when ready
```

oznaczający prośbę o zmianę dyskietki na właściwą (zawierającą IBMBIO.COM i IBMDOS.COM) i o naciśnięcie jakiegokolwiek

klawisza (co spowoduje ponowną próbę uruchomienia systemu operacyjnego). Przez "jakikolwiek klawisz" rozumiany jest jeden z klawiszy znakowych (dotyczy to również dalszej części tej książki). Jeżeli oba zbiory zostały znalezione i zidentyfikowane, dyskowy program ładujący powoduje wczytanie do pamięci operacyjnej zbioru IBMBIO.COM, którego zadaniem jest wczytanie zbioru IBMDOS.COM, inicjalizacja przyłączonych urządzeń peryferyjnych, ładowanie drajwerów (sterowników programowych) urządzeń instalowanych przez użytkownika przy wykorzystaniu CONFIG.SYS, resetowanie (doprowadzenie do stanu początkowego) sterowników napędów dyskowych, ustawienie wektorów przerwań (tym nie będziemy się zajmować) itd. Ostatnią czynnością, wykonywaną przez oprogramowanie zawarte w IBMBIO.COM, jest relokowanie (przepisanie w pamięci RAM) IBMDOS.COM w dół do możliwie najniższego adresu i wywołanie tego programu. Z kolei najważniejszym zadaniem IBMDOS.COM jest przygotowanie miejsca w RAM dla COMMAND.COM oraz (przy wykorzystaniu wywołania funkcji EXEC) załadowanie i uruchomienie go. IBMDOS.COM wyświetla również numer wersji DOSu i komunikat o prawach autorskich twórców systemu. COMMAND.COM po przejęciu nadzoru nad sprzętem szuka zbioru AUTOEXEC.BAT i automatycznie realizuje polecenia, które są zawarte w tym zbiorze. Poleceniami tymi są programy działające pod nadzorem systemu operacyjnego (przeważnie są to polecenia systemowe). Jeżeli AUTOEXEC.BAT nie zostanie znaleziony, to na monitor jest wyprowadzany komunikat z prośbą o podanie aktualnego czasu i daty. Procesor poleceń COMMAND.COM może być zastąpiony przez dowolny inny procesor poleceń, np. stworzony przez użytkownika komputera na własne potrzeby. Nazwa nowego procesora poleceń musi zostać wyszczególniona jako parametr polecenia SHELL. W pamięci operacyjnej komputera prawie zawsze jest obecna część rezydentna programu COMMAND.COM, konieczna dla poprawnego działania całego systemu. Czasami jednak zostaje ona wymazana przez programy wywoływane przez użytkownika, dlatego niekiedy po wykonaniu uprzednio wywołanego programu system żąda włożenia dyskietki ze zbiorem COMMAND.COM w celu ponownego jego wczytania. Sytuacja taka nie występuje, jeżeli korzystamy z dysku twardego, wirtualnego lub dyskietki z zapisanym tam (i wskazywanym przez odpowiedni parametr systemo-

wy) zbiorem COMMAND.COM - ponowne załadowanie go do pamięci przebiega wtedy automatycznie.

Nasze wiadomości o rozpoczynaniu pracy przez system operacyjny i ogólnym sposobie jego działania możemy podsumować.

- ROM-BIOS kopiuje do pamięci dyskowy program ładujący i przekazuje mu nadzór.
- Dyskowy program ładujący sprawdza obecność zbiorów IBMBIO .COM i IBMDOS.COM na dysku, wczytuje je do pamięci i przekazuje nadzór zbiorowi IBMBIO.COM.
- IBMBIO.COM inicjalizuje pracę urządzeń zewnętrznych oraz uruchamia IBMDOS.COM.
- System szuka zbioru CONFIG.SYS i uwzględnia żądania użytkownika zawarte w tym zbiorze, a dotyczące konfiguracji i pewnych szczegółów pracy systemu.
- System wczytuje COMMAND.COM i przekazuje mu nadzór nad sprzętem.
- COMMAND.COM szuka zbioru AUTOEXEC.BAT i (jeżeli taki zbiór istnieje) wykonuje polecenia w nim zawarte. Jeżeli AUTOEXEC .BAT nie istnieje, system żąda podania daty i czasu.
- Zostaje wyświetlone zgłoszenie się systemu, po czym system przechodzi w stan oczekiwania na polecenia użytkownika.
- COMMAND.COM po odebraniu polecenia szuka odpowiadającego mu polecenia wewnętrznego i jeżeli takie znajdzie, to wykonuje je natychmiast.
- Jeżeli polecenie użytkownika nie jest wewnętrzne, to system szuka programu o odpowiedniej nazwie (przypominam o priorytetach rozszerzeń nazw zbiorów programowych!) i wykonuje go. Po wykonaniu programu następuje ponowne zgjoszenie się systemu i oczekiwanie na kolejne polecenie.
- Jeżeli system nie może znaleźć żadnego wykonywalnego programu o nazwie zgodnej z poleceniem użytkownika, jest wyświetlany komunikat o błędzie i następuje ponowne przejście w stan oczekiwania na polecenie.

Można powiedzieć, że podstawowymi zadaniami systemu operacyjnego są:

- wykonywanie poleceń (programów),
- instalowanie i obsługa urządzeń zewnętrznych (pośredniczenie między sprzętem a programami i użytkownikiem),
- zarządzanie zbiorami programów i danych (zarządzanie zewnętrznymi pamięciami masowymi),
- zarządzanie pamięcią operacyjną.

Najpierw powiedzmy sobie kilka słów o sposobie organizacji zarządzania zgromadzoną w systemie komputerowym informacją (czyli o zarządzaniu zbiorami). Jak wiadomo, każdy system komputerowy służy do przetwarzania informacji. Ściśle rzecz biorąc, w procesie przetwarzania wielkością wejściową są dane, a wyjściową - informacja, którą pragnęliśmy uzyskać. Dla naszych potrzeb proponuję przyjąć konwencję nazywania danych wejściowych, wyjściowych i informacji po prostu danymi (dane wyjściowe jednego algorytmu przetwarzania mogą być danymi wejściowymi innego algorytmu). Dane zapisane w pamięci masowej komputera (pamięć taśmowa, dyskietki, twardy dysk, dysk optyczny itp.) są przechowywane w określonych strukturach, zwanych zbiorami (ang. file). Używane jest też bardzo często pojęcie pliku, będące niewątpliwie bardziej dokładnym tłumaczeniem słowa file. Ponieważ jednak w polskiej literaturze przez plik coraz częściej określana jest pewna abstrakcja programowa, wiązana programowo z fizyczną strukturą na dysku, dla określenia tej struktury będziemy używać pojęcia zbiór. Zbiór danych można porównać z kartkami zszytymi z jednej strony - kartki te możemy przeglądać kolejno jedną po drugiej (sekwencyjnie), ale nie możemy zobaczyć wszystkich kartek jednocześnie. Zarazem jest możliwe przełożenie całego zbioru kartek z jednego miejsca w inne, zmiana tytułu, skopiowanie na kserografie. Tak samo jest ze zbiorami danych. Zbiór taki może być kopiowany, można zmieniać jego nazwę, może być przeglądany przez inny program "kartka po kartce", z możliwością kartkowania (przewijania zbioru) do końca lub na początek. Taki zbiór nazywa się zbiorem o dostępie sekwencyjnym. Zbiorem o dotępie bezpośrednim nazywamy zbiór, w którym poszczególne pola danych (kartki) są ponumerowane. Numeracja taka umożliwia skoki do wskazanych pól bez konieczności przewijania zbioru od początku do określonego miejsca, jest też możliwe przepisanie (przeniesienie w inne miejsce) kartek o

określonych numerach. Każdy zbiór, także programowy (czyli zawierający informację sterującą pracą komputera) nosi swoją nazwę. Nazwa może składać się z od jednej do ośmiu liter oraz jedno-, dwu- lub trzyliterowego jej rozszerzenia (np. GR.COM, KUKUKUKU.EXE, CO.H itp.). Ani w nazwie zbioru, ani w jej rozszerzeniu nie mogą występować znaki ASCII (ich tabela załączona jest na końcu książki) o kodzie dziesiętnym niższym i równym 32D (20H). Nie dopuszcza się również użycia następujących znaków:

" \ / [ ] : < > + = ; ,

Trzeba pamiętać o tym, że użytkownik nie może nazywać swoich zbiorów żadną z nazw zarezerwowanych przez system operacyjny dla określenia urządzeń wejścia/wyjścia. Bardzo często są stosowane pewne standardowe rozszerzenia nazw zbiorów. Takie standardowe rozszerzenie jest informacją dla użytkownika o rodzaju zbioru. Stosowanie określonych rozszerzeń do określonych rodzajów zbiorów z reguły nie jest obligatoryjne, chociaż bywa wielce pomocne i znacznie ułatwia wykonywanie różnych operacji na grupach zbiorów powiązanych ze sobą. Rozszerzenia mające szczególne znaczenie dla systemu operacyjnego to:

BAT (od ang. batch) - oznacza zbiór stworzony przez użytkownika i składający się z szeregu wywołań poleceń DOSu i programów, które system operacyjny może wykonać.

COM (od ang. command) - oznacza polecenie DOSu lub zbiór programowy, który może być wykonany przez DOS.

EXE (od ang. execute) - oznacza zbiór programowy, który może być wykonany przez DOS (istnieją pewne, dla nas nieistotne, różnice między zbiorami typu COM a EXE).

SYS (od ang. system) - oznacza zbiór zawierający informację o żądanej konfiguracji systemu (konkretnie jest to CONFIG .SYS) lub będący programowym sterownikiem jednego z instalowalnych urządzeń zewnętrznych.

$$$ oznacza zbiór uszkodzony lub "podejrzany" o pewne nieprawidłowości w zapisie.

Poniżej zostało podanych kilka najczęściej używanych rozszerzeń i ich znaczenie:

BAK (od ang. backup) - oznacza poprzednią wersję zbioru, który został zmieniony; rozszerzenie takie jest generowane automatycznie przez wiele programów.

C oznacza zbiór zawierający tekst źródłowy programu w języku C.

DAT lub DTA (od ang. data) - oznacza zbiór z danymi.

DOC (od ang. document) - oznacza zbiór tekstowy będący dokumentacją (opisem) innego zbioru. FTN (od nazwy języka programowania Fortran) - oznacza zbiór zawierający tekst programu w Fortranie.

HLP (od ang. help) - oznacza zbiór zawierający instrukcje, np. obsługi określonego programu.

LIB (od ang. library) - oznacza zbiór (katalog) biblioteczny, czyli zawierający informacje wykorzystywane przez konsolidatory.

MSG (od ang. message) - oznacza zbiór zawierający komunikaty programu dla użytkownika, czyli po prostu informację tekstową (podobnie, jak zbiory z rozszerzeniami DOC i HLP).

OVL (od ang. overlay) - oznacza zbiór będący tzw. nakładką, czyli zbiorem programowym używanym przez inny zbiór programowy.

PAS (od nazwy języka programowania Pascal) - oznacza zbiór zawierający tekst źródłowy programu w Pascalu.

TMP (od ang. temporary) - oznacza zbiór pomocniczy, generowany tymczasowo.

TXT (od ang. text) - oznacza zbiór tekstowy (zawierający tekst).

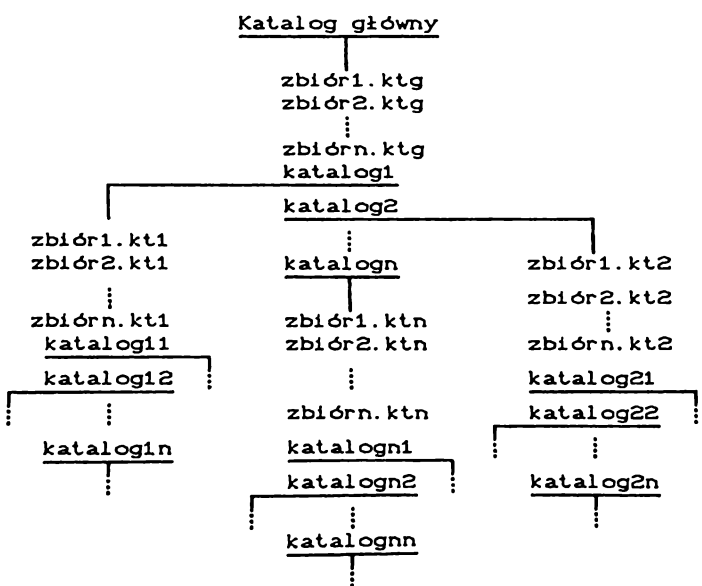
Dzięki nazwom oraz pewnym dodatkowym informacjom związanym z każdym zbiorem (o czym nieco później), zbiory stanowią dla komputera oddzielne, rozróżnialne obiekty. Ponieważ obecnie używane pamięci masowe mają olbrzymie pojemności i jest możliwe zapisanie tysięcy oddzielnych zbiorów na pojedynczym twardym dysku, w celu ułatwienia zarządzania zgromadzonymi danymi i programami wprowadzono pewne struktury zapisu. System operacyjny PC DOS operuje zbiorami i katalogami. Wiemy już z grubsza, czym jest zbiór. Katalog natomiast jest zbiorem zbiorów zgrupowanych pod wspólną nazwą (nazwy katalogów konstruuje się na tych samych zasadach, co nazwy zbiorów). Elementem każdego katalogu może być katalog niższego rzędu,

czyli podkatalog katalogu nadrzędnego. Katalogi wraz z podkatalogami tworzą na dyskach struktury nazywane drzewami (ang. tree-structured directories). Na każdym dysku znajduje się katalog główny (ang. root directory). Może on w sobie zawierać różne zbiory i podkatalogi (ang. subdirectory) zdefiniowane przez użytkownika. Liczba nazw zbiorów i katalogów w katalogu głównym nie jest nieograniczona - zależy ona od typu używanego napędu dyskowego. Obszar przeznaczony na nazwy na dyskietce jednostronnej może pomieścić 64 pozycje (ang. directory entries), a na najpopularniejszych dyskietkach dwustronnych 360 KB może zawierać do 112 pozycji. Dyskietki o poczwórnej (wysokiej) gęstości zapisu i pojemności 1.2 MB mają w katalogu głównym miejsce na 224 nazwy. Katalog główny twardego dysku (niezależnie od jego pojemności) pozwala na użycie 512 nazw zbiorów i katalogów. Trzeba wyraźnie powiedzieć, że te ograniczenia ilościowe dotyczące liczby użytych nazw zbiorów i katalogów odnoszą się tylko do katalogu głównego. Każdy katalog niższego rzędu może zawierać zupełnie dowolną ilość nazw innych zbiorów i katalogów. Dzięki takim możliwościom można (szczególnie na twardych dyskach) tworzyć drzewa katalogów bardzo rozbudowane w dół. Przestrzegałbym jednak przed dopuszczeniem do nadmiernego "rozrastania" się struktury drzewa - uzyskanie dostępu do zbiorów zawartych w katalogach najniższych poziomów (katalog główny to poziom najwyższy) pochłania sporo czasu. Jest to spowodowane koniecznością przejścia do tego katalogu, z którego chcemy korzystać. DOS w każdej chwili widzi tylko jeden katalog - katalog aktualny (ang. current directory) na aktualnym dysku (ang. default drive). Zaraz po włączeniu komputera aktualnym napędem staje się ten napęd, z którego został wczytany system operacyjny, a aktualnym katalogiem jest katalog główny na napędzie aktualnym. Jeżeli przyjrzymy się bliżej konstrukcji katalogów i roli odgrywanej przez system operacyjny w zarządzaniu zbiorami, to nasunie nam się porównanie z mieszkaniem, pokojami i człowiekiem. Spróbujmy wyobrazić sobie, że nasz system operacyjny to człowiek, poszczególne napędy dysków to mieszkania a katalogi to pokoje. Wtedy jasnym staje się, że człowiek-system operacyjny w danej chwili może przebywać tylko w jednym mieszkaniu-dysku oraz znajdować się tylko w jed-

nym z pokoi-katalogów. Jeżeli sprzęty i przedmioty w mieszkaniach porównamy do zbiorów na dysku, to katalog główny staje się korytarzem, w którym znajduje się kilka sprzętów i wejścia do innych pokoi. Ale nie do wszystkich - istnieją przecież pokoje, do których można przedostać się tylko przechodząc przez inne pokoje. W takim mieszkaniu mamy możliwość niemal dowólnego dobudowywania i wyburzania pokoi (tworzenia i usuwania katalogów). Trzeba tylko trzymać się pewnych zasad. Pierwsza z nich: w wyburzanym pokoju (usuwanym katalogu) nie mogą znajdować się żadne sprzęty (wszystkie zbiory powinny być uprzednio usunięte). Druga: dobudowanie nowego pokoju (stworzenie nowego katalogu) powinno być dobrze przemyślane i uzasadnione. Podczas pracy z katalogami należy pamiętać, że co prawda system operacyjny zawsze "widzi" tylko katalog aktualny na dysku aktualnym, ale na każdej dyskietce (dysku) zawsze jest otwarty ten katalog, który ostatnio był katalogiem aktualnym. Przeanalizujmy następujący przykład: jeżeli pracujemy na twardym dysku C: w katalogu drugiego poziomu \SYSUT\KOPIKI, a poprzednio pracowaliśmy na napędzie A: w katalogu \TOOLBOX, to po przejściu z dysku C: na A: znajdziemy się od razu w katalogu \TOOLBOX, a nie w katalogu głównym \ (root directory). Analogicznie, po zmianie aktualnego dysku z A: na C: znajdziemy się w \SYSUT\KOPIKI. Organizowanie zbiorów w katalogi wymaga także określania tzw. ścieżek dostępu (ang. path). Upraszczając, można powiedzieć, że ścieżka dostępu zawiera informację o katalogach, w których system operacyjny ma szukać programu do wykonania. Sam system operacyjny po ścieżkach szuka tylko zbiorów z rozszerzeniami COM, EXE i BAT. Natomiast wiele programów użytkowych wymaga podania ścieżki dostępu do innych zbiorów, z których korzystają (np. zbiory zawierające dane do przetworzenia, zbiory wynikowe, nakładki), a które znajdują się w innych katalogach. Takimi programami są m.in. kompilator języka Pascal firmy Borland (popularny Turbo Pascal), edytory tekstów ChiWriter czy WordStar. Nieco starsze programy, pisane pod wcześniejsze wersje DOSu (nie dające możliwości budowania katalogów), mogą sprawić użytkownikowi wiele kłopotów. Z reguły programy takie należy kopiować do katalogu głównego i stamtąd je uruchamiać, a w razie niepowodzenia można też próbować różnych (nie zaw-

sze bezpiecznych) zabiegów ze specjalnymi poleceniami DOSu. Dotyczy to właściwie programów bardziej rozbudowanych, korzystających z innych zbiorów lub zapisujących coś na dysk. Znana gra DigDug zawsze usiłuje zapisywać wyniki na dyskietkę znajdującą się w stacji A:, nawet w przypadku uruchomienia tej gry z twardego dysku. Przyczyną takich kłopotów jest zapisanie w programie na stałe miejsca poszukiwania programów współpracujących lub wpisywania wyników. Brak możliwości zmiany tej informacji powoduje nieprawidłowe działanie programu w niektórych środowiskach pracy.

Bardziej szczegółowe informacje dotyczące konkretnych poleceń systemu operacyjnego zostaną podane w następnym rozdziale. Przykład struktury drzewa katalogów jest przedstawiony poniżej (nazwy katalogów zostały podkreślone). Jak łatwo zauważyć, struktura ta rzeczywiście przypomina drzewo, tyle tylko, że rosnące w dół a nie w górę. Korzeń, czyli katalog główny, znajduje się najwyżej. Kolejne katalogi są budowane w dół.



Struktura drzewa katalogów.

Wiemy już, w jaki sposób jest uruchamiany system operacyjny, do czego służy, jak jest zorganizowane przechowywanie danych. Teraz dowiemy się, jakie podstawowe czynności mogą być wykonane przez użytkownika za pośrednictwem systemu operacyjnego. Trzeba bowiem zdać sobie sprawę z tego, że tak naprawdę DOS (oprócz procesu inicjalizacji pracy systemu) sam z siebie nic nie robi. Czeka tylko na polecenia użytkownika. Polecenia te mogą dotyczyć następujących, podstawowych czynności:

- zmiana dysku aktualnego,
- sprawdzenie poprawności zapisu danych na dysku (polecenie CHKDSK),
- czyszczenie ekranu (polecenie CLS),
- porównanie zawartości dwóch zbiorów (polecenie COMP),
- zmiana katalogu aktualnego (polecenie CHDIR, czyli CD)
- skopiowanie zbioru lub grupy zbiorów (polecenie COPY),
- ustawienie daty i czasu (polecenia DATE i TIME),
- wyświetlenie nazw zbiorów i katalogów (oraz informacji z nimi związanej) znajdujących się na dysku (polecenie DIR),
- porównanie zawartości dwóch dyskietek i wyświetlenie wyniku operacji (polecenie DISKCOMP),
- kasowanie zbiorów (polecenie ERASE lub DELETE),
- tworzenie partycji DOSu (części używanej przez DOS) na twardym dysku podczas jego formatowania (polecenie FDISK),
- znajdowanie łańcuchów tekstowych (ciągów znaków) w określonym zbiorze (polecenie FIND),
- formatowanie dysków (polecenie FORMAT),
- tworzenie katalogów na dyskach (polecenie MKDIR, czyli MD),
- tworzenie, usuwanie i podawanie informacji o ścieżkach dostępu (polecenie PATH),
- wysyłanie zbiorów na drukarkę (polecenie PRINT),
- zmienianie nazw istniejących zbiorów (polecenie RENAME, czyli REN),
- usuwanie pustych katalogów z dysku (polecenie RMDIR, czyli RD),
- sortowanie zawartości zbiorów (polecenie SORT),
- podawanie informacji o katalogach istniejących na dysku i ścieżkach dostępu do nich (polecenie TREE),
- wyświetlanie zawartości zbiorów na ekranie monitora (polecenie TYPE),

- podanie numeru wersji DOSu (polecenie VER).
- sprawdzanie poprawności zapisu danych na dysku (polecenie VERIFY),
- nadawanie nazw dyskom (polecenie VOL),
- kierowanie wyniku wykonania danego polecenia na wejście innego polecenia (znak :),
- wysyłanie danych lub wyniku wykonania danego polecenia do wskazanego urządzenia lub zbioru (znak >),
- pobieranie danych ze wskazane go zbioru do wykonania wskazanego polecenia (znak <),
- dopisywanie wyniku działania danego polecenia na koniec już istniejącego zbioru (znaki >>).

Przedstawione tutaj w wielkim skrócie funkcje DOSu są w rzeczywistości znacznie obszerniejsze. Niektóre polecenia dają nam bardzo duże możliwości przetwarzania danych i efektywnej obsługi sprzętu. Często głównym ograniczeniem działania nie jest niedostatek funkcji DOSu, lecz brak wyobraźni (lub wiedzy) użytkownika. Trzeba jednak przyznać, że obsługa systemu operacyjnego niekiedy może być pracochłonna - szczególnie przy wpisywaniu wszystkich poleceń, opcji i nazw urządzeń z klawiatury. Dlatego w samym systemie operacyjnym istnieje wiele mechanizmów pozwalających na znaczne ułatwienie sobie pracy i pewne "zautomatyzowanie" codziennie wykonywanych operacji. W rozdziale czwartym zostanie przytoczonych kilka przykładów takiej automatyzacji DOSu, ułatwiającej wykonywanie standardowych czynności związanych z obsługą sprzętu i pozwalającej na oszczędzanie czasu, klawiatury, palców i nerwów. W rozdziale piątym zostanie omówionych kilka programów dublujących czynności wykonywane zwykle za pośrednictwem systemu operacyjnego i pozwalających na realizowanie pewnych zleceń za pomocą naciśnięcia pojedynczego klawisza. Zanim jednak zagłębimy się w lekturze rozdziałów czwartego i piątego, czeka nas rozdział trzeci, prowadzący Czytelnika za rękę podczas pierwszej sesji z komputerem i omawiający polecenia DOSu (wersja 3.10).

3. SAM NA SAM Z DOSEM

Zacznijmy nasze "sam na sam z DOSem" od nauki operacji, które są wykonywane podczas uruchamiania systemu po raz pierwszy. Ze względu na różne konfiguracje sprzętu część informacji zawartych w tym rozdziale kierowana będzie do posiadaczy określonego zestawu urządzeń. Jest oczywiste, że każdy ma jednostkę centralną, monitor i klawiaturę - ale nie wszyscy są właścicielami (czy też użytkownikami) systemów z twardym dyskiem lub dwoma napędami miękkich dysków. Proponuję zatem wyróżnić trzy podstawowe konfiguracje sprzętowe i nazwać je odpowiednio:

- system SFDD (ang. single floppy disk drive system), czyli konfiguracja z jednym napędem miękkich dysków,
- system DFDD (ang. double floppy disk drive system), czyli konfiguracja z dwoma napędami miękkich dysków,
- system HDD (ang. hard disk drive system), czyli konfiguracja z napędem twardego dysku i z jednym lub dwoma napędami miękkich dysków.

Nie ma wielkiej różnicy między konfiguracjami SFDD a DFDD. Jeżeli mamy sprzęt w konfiguracji SFDD, to system operacyjny łatwo symuluje pracę z dwoma napędami miękkich dysków. W zależności od sytuacji jeden napęd istniejący w systemie traktowany jest albo jako napęd A:, albo jako napęd B:. Użytkownik proszony jest przez DOS o włożenie do napędu tej dyskietki, która powinna w danej chwili znaleźć się w napędzie o danej nazwie. Dlatego ograniczę się do opisu różnic między pracą z komputerem w konfiguracji DFDD a HDD. Większość czynnoś-

ci związanych z obsługą systemu jest taka sama w obydwu przypadkach (np. będziemy często używać określenia "napęd aktualny" w odniesieniu tak do twardego dysku, jak i do napędów miękkich dysków), ale w razie wystąpienia istotnych różnic użytkownicy SFDD będą mogli znaleźć informacje ich dotyczące pod wierszem z napisem *SFDD:*, a użytkownicy HDD odpowiednio pod wierszem z napisem *HDD:*. Wydzielone akapity dotyczące określonej konfiguracji sprzętowej będą zakończone symbolem □.

3.1 Uruchomienie systemu

Jak zostało wspomniane na wstępie, naszą naukę posługiwania się systemem operacyjnym PC-DOS rozpoczniemy od sesji z komputerem uruchamianym po raz pierwszy. Po włączeniu komputera do sieci (pamiętajmy o odpowiedniej kolejności włączania sprzętu!) wkładamy dyskietkę z zapisanym na niej systemem operacyjnym do napędu A: (przeważnie jest to górny napęd) i opuszczamy w dół dźwignię znajdującą się na płycie czołowej obudowy napędu (tzw. drzwi napędu). Po wczytaniu się system zażąda podania daty:

```
Current date is Tue  1-01-1980
Enter new date (mm-dd-yy):
```

czyli

```
Bieżąca data Wtorek  1-01-1980
Podaj nową datę (mm-dd-rr):
```

Datę podajemy w sposób podpowiedziany nam przez system, tzn. dwie cyfry oznaczające miesiąc, myślnik, dwie cyfry oznaczające dzień miesiąca, myślnik, dwie cyfry oznaczające rok. Zamiast myślnika możemy użyć kropki . lub znaku / (ang. slash). Jeżeli podczas wprowadzania daty zdarzy nam się pomylić, to do usunięcia niewłaściwego znaku (lub całego ich ciągu) służy klawisz Backspace, oznaczony strzałką skierowaną w lewo i umieszczany przeważnie w prawym górnym rogu głównego bloku klawiatury (nad klawiszem ENTER). Klawisz ten usuwa znak na lewo od kursora. Po napisaniu daty wciskamy klawisz ENTER

(wciśnięcie tego klawisza będzie często oznaczane przez symbol <ENTER>), co spowoduje wczytanie napisanego tekstu przez system operacyjny. Następnie system zażąda podania godziny (czas wyświetlony na końcu pierwszej linii komunikatu jest czasem, jaki upłynął od momentu włączenia komputera do momentu wyświetlenia tego komunikatu):

```
Current time is  0:00:52.87
Enter new time:
```

czyli

```
Bieżący czas  0:00:52.87
Podaj nowy czas:
```

Czas podajemy, wpisując godziny, minuty i sekundy rozdzielone dwukropkami, np. 12:34:30. Minut i sekund nie trzeba koniecznie podawać - jeżeli nie napiszemy nic w miejscu przewidzianym na podanie minut i sekund, to system sam ustawi je na zero. Czas przez nas podany będzie odmierzany dopiero po wciśnięciu klawisza ENTER. Jeżeli nie zależy nam na aktualizowaniu daty i czasu systemu, możemy nie podawać żadnych wartości, tylko dwukrotnie wcisnąć klawisz ENTER. Zegar systemowy pozostanie wtedy ustawiony na datę i godzinę uprzednio wyświetlone.

HDD:
Twardy dysk z reguły jest dostarczany już z kopią systemu operacyjnego. Jeżeli nie mamy osobnej dyskietki z uprzednio skopiowanym DOSem, to po włączeniu komputera nie trzeba nic robić - system zostanie automatycznie wczytany z twardego dysku. Komunikaty wyprowadzane przez DOS na monitor będą wyglądać jak wyżej pod warunkiem, że na dysku nie istnieje zbiór AUTOEXEC.BAT.
□

Po wczytaniu daty i czasu system zgłasza się znakiem gotowości, który składa się standardowo z oznaczenia aktualnego napędu (np. A) i znaku zachęty (>):

```
A>
```

Jeżeli na dysku systemowym istnieje zbiór AUTOEXEC.BAT, to system najpierw automatycznie wykona polecenia zawarte w AUTOEXEC.BAT, a potem pojawi się znak zgłoszenia.
Pamiętajmy, że bezpośrednio po uruchomieniu systemu napędem aktualnym jest ten napęd, z którego system został wczytany.

3.2 Kopiowanie dyskietek i zbiorów

Po pierwszym zgłoszeniu się systemu najważniejszą czynnością będzie sporządzenie jego kopii roboczej. W tym celu wykorzystamy polecenie **FORMAT**. Polecenie to przygotowuje dyskietkę do operacji zapisania danych. Podczas formatowania dyskietki uprzednio używanej są tracone wszelkie dane wcześniej na niej zapisane. Dlatego polecenie **FORMAT** powinno być używane bardzo ostrożnie - szczególnie sformatowanie twardego dysku i stracenie z niego wszystkich danych może okazać się przykre. Dyskietki, które nie zostały sformatowane nie mogą służyć do zapisu i odczytu danych - przed pierwszym użyciem dyskietki jako nośnika informacji należy ją sformatować. Pamiętajmy też o tym, żeby w napędach 1.2 MB używać dyskietek HD (ang. high density - duża gęstość zapisu), a w napędach 360 KB dyskietek DD (ang. double density - podwójna gęstość zapisu). Nie jest też wskazane używanie dyskietek jednostronnych w napędach dwugłowicowych (obecnie niemal wszystkie napędy miękkich dysków są napędami dwugłowicowymi). Druga strona dyskietki jednostronnej nie jest fabrycznie testowana i może nie być pełnowartościowa. Niekiedy oszczędzenie kilku złotych na dyskietkach może spowodować niewymierne straty wielu kilobajtów danych i wielu godzin pracy.
FORMAT nie jest poleceniem wewnętrznym - na dyskietce w napędzie aktualnym musi zatem znajdować się zbiór o nazwie FORMAT.COM.
Po znaku zgłoszenia się systemu piszemy zatem:

```
A>format b:/s/v
```

gdzie:
- B: jest oznaczeniem napędu, w którym ma nastąpić formatowanie,

- użycie parametru /s powoduje przeniesienie zbiorów systemowych (IBMBIO.COM, IBMDOS.COM i COMMAND.COM z dyskietki, z której zostało wywołane polecenie **FORMAT** na dyskietkę formatowaną,
- użycie parametru /v pozwala na nadanie nazwy formatowanej dyskietce.

Inne parametry polecenia **FORMAT** zostaną omówione później.
Po wciśnięciu klawisza ENTER system odpowie komunikatem:

```
Insert new diskette for drive B:
and strike ENTER when ready
```

czyli

```
Włóż nową dyskietkę do napędu B:
i wciśnij ENTER, gdy będziesz gotowy
```

Po włożeniu nowej dyskietki do napędu B: wciskamy ENTER, co rozpoczyna proces formatowania, przeniesienia zbiorów systemowych i nadania nazwy dyskietce. Podczas formatowania system informuje nas o przebiegu tej operacji (wyprowadzenie komunikatu - Head: *nr głowicy* Cylinder: *nr ścieżki*). Po zakończeniu formatowania pojawia się komunikat:

```
Format complete
```

czyli

```
Formatowanie zakończone
```

Następnie zostają przeniesione zbiory systemowe, o czym DOS znowu nas informuje:

```
System transferred
```

czyli

```
System przeniesiony
```

Ostatni krok to nadanie dyskietce nazwy - zostaje wyświetlony komunikat:

```
Volume label (11 characters, ENTER for none)?
```

czyli

```
Nazwa dyskietki (11 znaków, wsiśnięcie tylko ENTER nie
nadaje nazwy)?
```

Teraz możemy z klawiatury wprowadzić nazwę dyskietki - piszemy np. SYSTEM i wciskamy ENTER. Można też pozostawić dyskietkę bez nazwy - wciskamy ENTER nic nie pisząc. Stosowanie nazw dyskietek jest jednak bardzo zalecane - później znacznie ułatwia to utrzymanie porządku w zgromadzonych danych. Po wprowadzeniu przez nas nazwy system poda informację o stanie dyskietki i zapyta się, czy chcemy sformatować następną dyskietkę:

```
362496 bytes total disk space
 69632 bytes used by system
292864 bytes available on disk

Format another (Y/N)?
```

czyli

```
362496 bajtów na dyskietce
 69632 bajtów użytych przez system
292864 bajtów do dyspozycji

Sformatować następną dyskietkę (T/N)?
```

Liczby występujące w komunikacie mogą być różne dla różnych wersji systemu operacyjnego i dla różnych typów napędów. Liczby podane powyżej odnoszą się do DOSu 3.20 na dyskietce sformatowanej w napędzie 360 KB. System pyta użytkownika, czy chce on sformatować następną dyskietkę? Ponieważ na razie nie chcemy tego robić, wciskamy najpierw klawisz z literą n, a potem ENTER. Na monitorze pojawi się znak zgłoszenia się systemu:

```
A>
```

HDD:
Przestrzegam przed przypadkowym sformatowaniem twardego dysku! Jeżeli w poleceniu **FORMAT** nie podamy nazwy stacji, w której ma nastąpić formatowanie, a stacją aktualną jest C: (twardy dysk), to po wciśnięciu ENTER rozpocznie się formatowanie dysku C: (aktualnego). Spowoduje to zniszczenie wszystkich danych zgromadzonych na tym dysku.
□

Należy dodać jeszcze, że użycie polecenia **FORMAT** w ogóle bez żadnych parametrów powoduje sformatowanie dyskietki w napędzie aktualnym (lub twardego dysku, jeżeli jest on napędem aktualnym). System nie jest przenoszony.
Jak już wiemy, formatowanie jest poprzedzone wyprowadzeniem przez system komunikatu z prośbą o włożenie do napędu nowej dyskietki (przyjmujemy, że napęd A: jest napędem aktualnym):

```
Insert new diskette for drive A:
and strike ENTER when ready
```

czyli

```
Włóż nową dyskietkę do napędu A:
i wciśnij ENTER, gdy będziesz gotowy
```

Mimo to jest dozwolone formatowanie dyskietki wcześniej sformatowanej i używanej. Operacja ponownego formatowania może być przydatna, jeżeli uprzednio występowały kłopoty z zapisem danych na dyskietce. Tak zwane przeformatowanie niekiedy pomaga uratować dyskietkę, ale trzeba pamiętać, że na dyskietce "z odzysku" nie powinno się trzymać danych szczególnie ważnych.
Gdy już uporaliśmy się z formatowaniem, możemy sprawdzić, której wersji DOSu używamy. W tym celu posłużymy się poleceniem **VER**:

```
A>ver <ENTER>
```

System na podane polecenie odpowie następującym komunikatem:

```
IBM Personal Computer DOS Version 3.10
```

Oczywiście, numer wersji DOSu Czytelnika może być inny.
W celu pełnego przepisania DOSu na naszą tworzoną kopię roboczą systemu należy z oryginalnej dyskietki systemowej przenieść wszystkie zbiory używane przez system operacyjny. Zbiory te są przeważnie programami realizującymi zewnętrzne polecenia DOSu. Niektóre spośród zbiorów systemowych służą jako programowe sterowniki urządzeń instalowanych przez użytkownika i wykorzystywane są przez DOS podczas konfigurowania systemu bezpośrednio po uruchomieniu go. Do skopiowania zbiorów systemowych użyjemy wewnętrznego polecenia **COPY** (użytkownicy wersji 3.20 mogą podać zewnętrzne polecenie **XCOPY**) w postaci:

```
A>copy a:*.* b: <ENTER>
```

lub w postaci uproszczonej

```
A>copy a: b: <ENTER>
```

Wszystkie zbiory z dyskietki w napędzie A: zostaną kolejno skopiowane na dyskietkę w napędzie B:. Użycie polecenia **XCOPY**:

```
A>xcopy a:*.* b: <ENTER>
```

spowoduje najpierw wczytanie wszystkich zbiorów z dyskietki źródłowej A: do pamięci operacyjnej (tzn. tylu zbiorów, na ile pozwoli wielkość pamięci), a następnie zapisanie ich na dyskietkę docelową B:. Operacja taka jest szczególnie wygodna dla osób dysponujących tylko jednym napędem miękkich dysków - nie trzeba przekładać dyskietek podczas kopiowania kolejno zbioru po zbiorze, jak w przypadku zastosowania polecenia **COPY**. A oto lista zbiorów,które powinny zostać skopiowane z dyskietki systemowej:

```
ANSI     SYS
ASSIGN   COM
ATTRIB   EXE
BACKUP   COM
CHKDSK   COM
```

```
COMMAND  COM
COMP     COM
DISKCOMP COM
DISKCOPY COM
DRIVER   SYS
EDLIN    COM
FDISK    COM
FIND     EXE
FORMAT   COM
GRAFTABL COM
GRAPHICS COM
JOIN     EXE
LABEL    COM
MODE     COM
MORE     COM
PRINT    COM
RECOVER  COM
REPLACE  EXE
RESTORE  COM
SELECT   COM
SHARE    EXE
SHOWTIME EXE
SORT     EXE
SUBST    EXE
SYS      COM
TIMER    COM
TREE     COM
VDISK    SYS
XCOPY    EXE
```

Na dyskietce systemowej może znaleźć się więcej lub mniej zbiorów - nawet te same wersje systemów operacyjnych mogą nieco różnić się od siebie zawartością (w zależności od daty wydania). IBM i Microsoft nie są zbyt konsekwentne, jeżeli chodzi o zgodność zawartości oryginalnych dyskietek systemowych z dołączanymi do nich opisami. I tak na przykład niniejsza książka powstaje w edytorze tekstów ChiWriter, pracującym pod nadzorem systemu MS DOS 3.20 (nr seryjny 036014-031000 .320). Na firmowych dyskietkach z tym systemem znalazły się

takie zbiory, jak APPEND.EXE i FC.EXE, które powinny być dostarczane dopiero z DOSem 3.30. Wróćmy jednak do kopiowania zbiorów systemowych. Po wykonaniu kopiowania DOS ponownie zgłosi się znakiem gotowości. Do wyspecyfikowania wszystkich zbiorów na dyskietce użyliśmy znaku gwiazdka (*). Znak ten zastępuje dowolny ciąg znaków dozwolonych w nazwie i rozszerzeniu nazwy zbioru. Znak zapytania (?) zastępuje dowolny pojedynczy znak z nazwy zbioru lub z jej rozszerzenia. Znaki * i ? noszą nazwę znaków blankietowych (ang. wild characters). Jeżeli podamy polecenie

```
A>xcopy *.com b:
```

to wynikiem wykonania go będzie skopiowanie wszystkich zbiorów o dowolnej nazwie i rozszerzeniu COM z napędu aktualnego na napęd B:. Polecenie

```
A>xcopy mo??.??? b:
```

spowoduje skopiowanie wszystkich zbiorów o dwóch pierwszych literach nazwy MO, dwóch następnych dowolnych i dowolnym rozszerzeniu z napędu aktualnego na napęd B:. Takie polecenie nie jest równoważna poleceniu

```
A>xcopy mo*.* b:
```

bowiem MO?? dotyczy tylko i wyłącznie nazw o długości od dwóch do czterech liter, natomiast MO* odnosi się do nazw o długości od dwóch do ośmiu liter. Jeżeli chodzi o rozszerzenie nazwy, to oznaczenie ??? jest równoważne oznaczeniu * (obydwa oznaczenia dotyczą trzyliterowych rozszerzeń nazw zbiorów). Polecenie

```
A>xcopy mo*.?? <ENTER>
```

odnosi się do do zbiorów o dwóch pierwszych literach nazwy M (długość nazwy od dwóch do ośmiu liter), i jedno- lub dwuliterowym rozszerzeniu. Natomiast polecenie

```
A>xcopy zbr?tom.nr?
```

odnosi się do wszystkich zbiorów o trzech pierwszych literach nazwy ZBR, piątej dowolnej, trzech ostatnich TOM i o rozszerzeniu z dwiema pierwszymi literami nr i ostatniej dowolnej (może to być np. cyfra oznaczająca numer porządkowy zbioru). Dyskietkę można także skopiować używając polecenia **DISKCOPY**, które w przypadku kopiowania na dyskietkę nie sformatowaną automatycznie przeprowadzi formatowanie. Polecenie to może być używane w postaci

```
A>diskcopy a: b: <ENTER>
```

przez użytkowników sprzętu z dwoma napędami miękkich dysków po 360 KB, lub w postaci

```
A>diskcopy <ENTER>
```

przez użytkowników sprzętu z jednym z dwoma różnymi (1.2 MB i 360 KB) napędami miękkich dysków. Nie można za pomocą tego polecenia kopiować zawartości twardego dysku na dyskietki - polecenie służy jedynie do kopiowania dyskietek. Podczas kopiowania przez polecenie **DISKCOPY** jest fizycznie przepisywana cała zawartość dyskietki - oznacza to, że jeżeli na dyskietce źródłowej znajdują się ukryte zbiory systemowe, to zostaną one przeniesione na dyskietkę docelową wraz ze wszystkimi innymi zbiorami z dyskietki źródłowej. Polecenie to (tak, jak **FORMAT**) nie jest poleceniem wewnętrznym, zatem na dyskietce w napędzie aktualnym powinien znajdować się zbiór o nazwie DISKCOPY.COM.
Po podaniu polecenia:

```
A>diskcopy a: b: <ENTER>
```

system odpowie komunikatem

```
Insert SOURCE diskette in drive A:

Insert TARGET diskette in drive B:

Press any key when ready . . .
```

czyli

Włóż dyskietkę ŹRÓDŁOWĄ do napędu A:

Włóż dyskietkę DOCELOWĄ do napędu B:

Wciśnij dowolny klawisz, gdy będziesz gotowy . . .

Po włożeniu dyskietek w odpowiednie napędy i naciśnięciu jakiegokolwiek klawisza system poinformuje nas o wykonywanej operacji kopiowania:

Copying 40 Tracks
9 Sectors/Track, 2 Side(s)

czyli

Kopiowanie 40 ścieżek
9 sektorów na ścieżkę, 2 strony

Po wykonaniu kopiowania pojawi się komunikat:

Copy another diskette (Y/N)?

czyli

Skopiować inną dyskietkę (T/N)?

Jeżeli nie chcemy kopiować innej dyskietki, wciskamy klawisz n, a potem ENTER. Następnie pojawia się znak zgłoszenia systemu.
W przypadku podania polecenia

A>diskcopy <ENTER>

system odpowie komunikatem

Insert SOURCE diskette in drive A:

Press any key when ready . . .

Po włożeniu dyskietki źródłowej i wciśnięciu jakiegokolwiek klawisza pojawi się komunikat:

```
Copying 40 Tracks
9 Sectors/Track, 2 Side(s)
```

i po chwili

```
Insert TARGET diskette in drive A:

Press any key when ready
```

Zmieniamy dyskietkę ze źródłowej na docelową i wciskamy dowolny klawisz. System zapisze dyskietkę docelową i zapyta się:

```
Copy another diskette (Y/N)?
```

Wciskamy n i ENTER.
Ilość kopiowanych ścieżek i sektorów podawana w komunikatach może być różna od wartości podanych w przykładach (jest to uzależnione od rodzaju napędu).

Jeżeli przy użyciu jednej stacji kopiujemy dyskietki o pojemności większej niż pamięć operacyjna naszego komputera (np. kopiowanie dyskietek 360 KB na maszynie z pamięcią 256 KB RAM), to operacja taka wymagać będzie zmian dyskietek źródłowej i docelowej w napędzie podczas kopiowania. Jest to spowodowane brakiem miejsca w pamięci operacyjnej komputera na przepisanie do niej całej zawartości dyskietki źródłowej - dyskietka jest kopiowana partiami. Podczas kopiowania system wyświetla komunikaty (już nam znane - Insert SOURCE diskette in drive A: i Insert TARGET diskette in drive A:) informujące użytkownika o konieczności zmiany dyskietki w napędzie.

3.3 Uruchomienie systemu z twardym dyskiem

Niekiedy początkujący użytkownik jest zmuszony do przygotowania twardego dysku do użytku. Nie zawsze bowiem twardy dysk

jest dostarczany przez sprzedawcę w stanie umożliwiającym natychmiastowe rozpoczęcie eksploatacji.
Twardy dysk można podzielić na 1÷4 części, nazywane partycjami. DOS 3.10 może jednorazowo zarządzać partycją do 32 MB, zatem dzielenie dysków o pojemnościach nie przekraczających 32 MB na partycje nie jest konieczne. Oprócz partycji DOSu, na twardym dysku mogą zostać wydzielone obszary do pracy pod nadzorem innych systemów operacyjnych. Do obsługi partycji służy polecenie FDISK. Aby je uruchomić, do stacji A: należy włożyć dyskietkę z systemem operacyjnym (i zbiorem FDISK .COMD, a następnie podać polecenie

```
A>fdisk <ENTER>
```

Na ekranie pojawi się menu, oferujące do wyboru 5 opcji:

```
1. Create DOS Partition
2. Change Active Partition
3. Delete DOS Partition
4. Display Partition Data
5. Select Next Fixed Disk Drive

Enter choice: [1]
```

czyli

```
1. Utworzenie partycji DOSu
2. Zmiana aktywnej partycji
3. Usunięcie partycji DOSu
4. Wyświetlenie informacji o partycji
5. Wybór innego twardego dysku

Wybierz opcję: [1]
```

Opcja 5 pojawia się tylko w przypadku konfiguracji sprzętu z więcej niż jednym twardym dyskiem.
Szczegółowo zajmiemy się pierwszą opcją menu. Po wciśnięciu klawisza ENTER (co spowoduje wybranie opcji 1), na ekranie pojawi się następny komunikat:

```
   Create DOS Partition

   Current Fixed Disk Drive: 1

   Do you wish to use the entire fixed disk
   for DOS (Y/N).....? [Y]
```

czyli

```
   Utworzenie partycji DOSu

   Aktualny twardy dysk: 1

   Czy chcesz przeznaczyć cały twardy dysk
   dla DOSu (T/N).....? [T]
```

Jeżeli naszym zamiarem jest przeznaczenie całego twardego dysku dla DOSu, to wciskamy Y (tak) - system odpowiada następującym komunikatem:

```
   System will now restart
   Insert DOS diskette in drive A:
   Press any key when ready . . .
```

czyli

```
   System uruchomi się ponownie
   Włóż dyskietkę systemową do stacji A:
   Wciśnij jakikolwiek klawisz, gdy będziesz gotowy . . .
```

Po upewnieniu się, że w stacji A: znajduje się dyskietka systemowa, naciskamy jakikolwiek klawisz. Po chwili na ekranie pojawią się zapytania o datę i czas. Gdy odpowiemy na nie, pojawi się znak zgłoszenia się systemu. Oznacza to, że partycja DOSu została przygotowana. Aby można było jej używać, należy przeprowadzić formatowanie przyporządkowanej jej części twardego dysku, podając polecenie

```
   A>format c:/s/v
```

Działanie takiego polecenie omówione jest (wraz ze sposobami przenoszenia całego systemu operacyjnego z dyskietki na dyskietkę) w częściach "Kopiowanie dyskietek i zbiorów" i "Polecenia DOSu" niniejszego rozdziału.
Jeżeli nie chcemy przeznaczyć całego twardego dysku na pojedynczą partycję DOSu, to w Create DOS Partition wciskamy N. FDISK poda wtedy informacje o wolnym miejscu na dysku i zażąda określenia rozmiaru tworzonej partycji. Partycja stworzona w ten sposób musi być aktywowana i sformatowana przed użyciem jej.
Jeżeli twardy dysk jest już gotowy do użytku, to wczytanie z niego systemu operacyjnego po zimnym lub gorącym starcie następuje wtedy, gdy:
- drzwi stacji A: są otwarte.
- na twardym dysku umieszczone zostały kopie zbiorów systemowych IBMBIO.COM, IBMDOS.COM i COMMAND.COM.
- partycja DOSu obejmuje cały dysk lub była aktywowana.

Systemu nie można uruchomić z drugiego twardego dysku (oznaczanego przez FDISK jako Fixed Disk Drive 2).

3.4 Podstawowe czynności

Po nauczeniu się formatowania i kopiowania dobrze byłoby posiąść umiejętność obejrzenia katalogu dyskietki. Podajmy polecenie

```
A>dir <ENTER>
```

Efektem jest lista zbiorów wyświetlana w sposób uniemożliwiający przeczytanie czegokolwiek. Spróbujmy dać to samo polecenie z parametrem /p (ang. page - strona):

```
A>dir/p <ENTER>
```

Ta postać polecenia DIR powoduje zatrzymanie wyświetlania listy zbiorów w momencie wypełnienia przez listę 24 wierszy na monitorze (wiersz 25 służy do wyświetlenia komunikatu). Po podaniu polecenia DIR/P ekran monitora powinien wyglądać mniej więcej tak:

```
 Volume in drive A is SYSTEM
 Directory of  A:

ANSI     SYS      1651   12-30-85   12:00p
ASSIGN   COM      1536   12-30-85   12:00p
ATTRIB   EXE      8247   12-30-85   12:00p
BACKUP   COM      6234   12-30-85   12:00p
BASIC    COM     19298   12-30-85   12:00p
BASICA   COM     36396   12-30-86   12:00p
CHKDSK   COM      9832   12-30-85   12:00p
COMMAND  COM     23791   12-30-85   12:00p
COMP     COM      4184   12-30-85   12:00p
DISKCOMP COM      5792   12-30-85   12:00p
DISKCOPY COM      6224   12-30-85   12:00p
DRIVER   SYS      1115   12-30-85   12:00p
EDLIN    COM      7508   12-30-85   12:00p
FDISK    COM      8173   12-30-85   12:00p
FIND     EXE      6416   12-30-85   12:00p
FORMAT   COM     11135   12-30-85   12:00p
GRAFTABL COM      1169   12-30-85   12:00p
GRAPHICS COM      3220   12-30-85   12:00p
JOIN     EXE      8955   12-30-85   12:00p
KEYBGR   COM      3278   12-30-85   12:00p
KEYBUK   COM      3036   12-30-85   12:00p
LABEL    COM      2346   12-30-85   12:00p
MODE     COM      6864   12-30-85   12:00p
Strike a key when ready . . .
```

Po wciśnięciu jakiegokolwiek klawisza wyświetlanie listy jest kontynuowane:

```
PRINT    COM      8976   12-30-85   12:00p
RECOVER  COM      4297   12-30-85   12:00p
REPLACE  EXE     11650   12-30-85   12:00p
RESTORE  COM      6012   12-30-85   12:00p
SELECT   COM      3826   12-30-85   12:00p
SHARE    EXE      8580   12-30-85   12:00p
SORT     EXE      1911   12-30-85   12:00p
SUBST    EXE      9911   12-30-85   12:00p
```

```
SYS      COM     4620  12-30-85  12:00p
TREE     COM     3357  12-30-85  12:00p
VDISK    SYS     3307  12-30-85  12:00p
XCOPY    EXE    11200  12-30-85  12:00p
       36 File(s)     33792 bytes free

A>
```

Rozpoczynając wyświetlanie listy zbiorów, system podał nam informacje o nazwie dyskietki i katalogu, do którego zbiory należą. Informacja o liczbie zbiorów w katalogu aktualnym i ilości wolnego miejsca (w bajtach) na dyskietce (dysku) została podana na końcu wyświetlonej listy. Do każdego z wyszczególnionych (wylistowanych - w gwarze informatycznej) zbiorów jest przypisana pewna informacja. Najpierw jest podawana nazwa zbioru, po przerwie (spacji) jej rozszerzenie, następnie wielkość zbioru w bajtach oraz data i godzina jego stworzenia (lub ostatniej zmiany jego zawartości). Jeżeli nie interesują nas te informacje i chcemy zobaczyć tylko nazwy zbiorów w katalogu aktualnym, możemy podać polecenie

```
A>dir/w <ENTER>
```

Parametr /w (ang. wide - szeroki) przy poleceniu DIR oznacza "szerokie" wyświetlanie listy zbiorów. Wygląda to następująco

```
 Volume in drive A is SYSTEM
 Directory of  A:

GRAFTABL COM  ASSIGN COM  ATTRIB   EXE  SORT EXE  BASIC  COM
BASICA   COM  CHKDSK COM  COMMAND  COM  COMP COM  FORMAT COM
DISKCOPY COM  DRIVER SYS  EDLIN    COM  FIND EXE  FDISK  COM
DISKCOMP COM  ANSI   SYS  GRAPHICS COM  JOIN EXE  KEYBGR COM
KEYBUK   COM  LABEL  COM  SELECT   COM  MORE COM  PRINT  COM
RECOVER  COM  XCOPY  EXE  RESTORE  COM  MODE COM  SHARE  EXE
BACKUP   COM  SUBST  EXE  SYS      COM  TREE COM  VDISK  SYS
REPLACE  EXE
       36 File(s)     33792 bytes free
```

Włóżmy teraz naszą świeżo utworzoną kopię roboczą DOSu do napędu A:, zamknijmy drzwi napędu i zresetujmy komputer. Rese-

towanie powoduje całkowite skasowanie pamięci operacyjnej maszyny i sprowadzenie urządzeń zewnętrznych do stanu początkowego sprzed wczytania systemu. Taki zabieg wykonuje się przez jednoczesne wciśnięcie klawiszy Ctrl (Control), Alt (Alternate) i Del (Delete) - jest to tzw. gorący start. Gorący start (w odróżnieniu od startu zimnego) omija procedury POST zawarte w BIOSie. Niektóre modele klonów mają dodatkowy przycisk umożliwiający przeprowadzenie resetowania sprzętowego (ang. hardware reset), czyli zimnego startu, przy zablokowanej klawiaturze. Zimny start jest praktycznie równoważny wyłączeniu i ponownemu włączeniu zasilania komputera. Jeżeli nasze nierozważne działania doprowadzą do tzw. zawieszenia systemu (brak reakcji na polecenia z klawiatury), to o ile używany komputer nie posiada przycisku uruchamiającego procedurę zimnego startu, jedynym ratunkiem jest wyłączenie sprzętu na kilkadziesiąt sekund i potem ponowne włączenie go. Po przeprowadzeniu resetowania system wczyta się ponownie z kopii roboczej.

3.5 Praca z katalogami

Sformatujmy teraz kolejną dyskietkę

```
A>format b:/s/v <ENTER>
```

którą wykorzystamy podczas poznawania operacji na katalogach. Po sformatowaniu przejdźmy na napęd B:

```
A>b: <ENTER>
```

i skorzystajmy z polecenia **MKDIR** (ang. make directory, czyli utwórz katalog), w skrócie **MD**. Spróbujmy utworzyć katalog o nazwie SYS:

```
B>md\sys <ENTER>
```

Pamiętajmy, że w katalogu głównym dyskietki B: są tylko trzy zbiory - widoczny COMMAND.COM i ukryte (posiadające atrybut hidden) IBMBIO.COM i IBMDOS.COM, oraz "wejście" (ang. direc-

tory entry) do nowo utworzonego katalogu \SYS. W katalogu \SYS nie ma żadnego zbioru, natomiast jest "wyjście" do katalogu głównego (będącego katalogiem nadrzędnym dla \SYS).

Do poruszania się po katalogach służy polecenie CHDIR (ang. change directory, czyli zmień katalog), w skrócie CD. Po podaniu polecenia

```
B>cd\sys <ENTER>
```

lub

```
B>cd sys <ENTER>
```

katalogiem aktualnym na napędzie aktualnym B: stanie się katalog \SYS. Znak \ (ang. backslash) każe systemowi szukać wskazanego katalogu jako podkatalogu katalogu głównego, natomiast spacja powoduje poszukiwanie w katalogu aktualnym. Przejdźmy ponownie na napęd A:

```
A>a: <ENTER>
```

i skopiujmy wszystki zbiory systemowe dyskietkę B:

```
A>copy *.* b: <ENTER>
```

Pamiętajmy, że na dyskietce B: był otwarty katalog \SYS, a nie katalog główny. Dlatego wszystkie zbiory z dyskietki A: zostaną skopiowane do katalogu \SYS. Taki sam skutek, jak COPY *.* B:, wywołałoby polecenie

```
B>copy a:*.* <ENTER>
```

podane z poziomu katalogu \SYS w B:.
Ponieważ do \SYS zostanie skopiowany także COMMAND.COM, którego poprzednia kopia (wykonana podczas przenoszenia systemu) znajduje się w katalogu głównym dyskietki B:, powinniśmy usunąć niepotrzebną kopię COMMAND.COM z katalogu \SYS. Do usunięcia tego zbioru wykorzystamy polecenie DEL (ang. delete, czyli usuwać):

```
A>del b:command.com <ENTER>
```

Polecenie DEL może odnosić się do całych grup zbiorów. Konstruowanie wzorów nazw dotyczących wybranych zbiorów odbywa się według tych samych reguł, które poznaliśmy przy omawianiu poleceń COPY i XCOPY.
Po usunięciu COMMAND.COM z katalogu \SYS przejdźmy ponownie na napęd B:

```
A>b: <ENTER>
```

i podajmy polecenie

```
B>dir/p <ENTER>
```

Na początku listy zbiorów zobaczymy tajemnicze kropki:

```
 Volume in drive B is SYSTEM
 Directory of  B:\SYS

              <DIR>      11-04-87   3:53p
              <DIR>      11-04-87   3:53p
ANSI     SYS     1651  12-30-85  12:00p
ASSIGN   COM     1536  12-30-85  12:00p
ATTRIB   EXE     8247  12-30-85  12:00p
itd.
```

Ciąg znaków <DIR> jest oznaczeniem specjalnego zbioru używanego przez system operacyjny do pracy z katalogami. Kropka pojedyncza (.) oznacza "wejście" do katalogu aktualnego. Dwie kropki (..) to "wyjście" do katalogu wyższego poziomu (ang. parent directory, czyli katalog-rodzic). Jeżeli katalogiem aktualnym jest SYS, to po podaniu polecenia

```
B>dir .. <ENTER>
```

zostanie wyświetlony spis zbiorów katalogu głównego (nadrzędnego w stosunku do \SYS). Polecenie

```
B>dir . <ENTER>
```

daje ten sam efekt, co polecenie

```
B>dir <ENTER>
```

Powrót do katalogu głównego umożliwi nam polecenie

```
B>cd\ <ENTER>
```

CD\ zawsze powoduje powrót do katalogu głównego z katalogu dowolnego poziomu. O tym, że rzeczywiście wróciliśmy do katalogu głównego, możemy przkonać się dając ponownie polecenie

```
B>dir <ENTER>
```

Na ekranie powinniśmy zobaczyć spis katalogu głównego:

```
 Volume in drive B is SYSTEM
 Directory of  B:

COMMAND  COM    23791  12-30-85  12:00p
SYS          <DIR>      11-04-87   3:53p
        2 File(s)     34164 bytes free
```

Ważną sprawą przy pracy z katalogami jest podawanie systemowi operacyjnemu ścieżek dostępu do zbiorów (czyli drogi, jaką system ma przebyć po katalogach w celu znalezienia zbioru). Wskazaniem ścieżki dostępu było podanie polecenia DEL B:COMMAND.COM, ponieważ określiliśmy miejsce, w którym ma znajdować się zbiór COMMAND.COM jako "katalog otwarty w napędzie B:. Teraz zapoznajmy się z następnym prostym przykładem wskazania ścieżki dostępu. Przejdźmy na napęd A: i z napędu B: wywołajmy zewnętrzne polecenie TREE w następujący sposób:

```
A>b:tree b: <ENTER>
```

Odpowiedzią na takie polecenie jest komunikat o błędzie, ponieważ wskazaliśmy systemowi operacyjnemu, że zbioru TREE.COM ma szukać w katalogu głównym (który jest katalogiem otwartym) dyskietki w napędzie B:. Tymczasem zbiór ten znajduje się w podkatalogu \SYS - dlatego nasze polecenie należy podać tak:

```
A>b:\sys\tree b: <ENTER>
```

Jeżeli napędem aktualnym byłby napęd B: a katalogiem aktualnym jego katalog główny, to ścieżka dostępu do polecenia **TREE** miałaby bardzo prostą postać:

```
B>\sys\tree b: <ENTER>
```

Polecenie **TREE** pokazuje nam wszystkie ścieżki dostępu na dyskietce (lub twardym dysku) we wskazanym napędzie. Użycie tego polecenia z parametrem /f (np. TREE C:/F) spowoduje wyświetlenie nie tylko ścieżek dostępu do wszystkich katalogów, ale i zbiorów znajdujących się w poszczególnych katalogach. Efekt działania podanego przez nas polecenia powinien być następujący:

```
DIRECTORY PATH LISTING FOR VOLUME SYSTEM
Path:\SYS
Sub-directories:  None
```

czyli

```
LISTA ŚCIEŻEK DOSTĘPU NA DYSKIETCE SYSTEM
Ścieżka:\SYS
Podkatalogi:  nie ma
```

Dla nieco bardziej rozbudowanego systemu katalogów na twardym dysku lista ścieżek może być taka:

```
DIRECTORY PATH LISTING FOR VOLUME TWARDZIEL
Path: \ACAD
Sub-directories:  None
Path: \CHI
Sub-directories:  CHITXT
Path: \CHI\CHITXT
Sub-directories:  None
Path: \SYMUL
Sub-directories:  MCAP
                  PSI
Path: \SYMUL\MCAP
Sub-directories:  NETS
Path: \SYMUL\MCAP\NETS
Sub-directories:  None
```

```
Path: \SYMUL\PSI
Sub-directories:  None
Path: \INNE
Sub-directories:  EU
                  FY
                  SKEY
Path: \INNE\EU
Sub-directories:  None
Path: \INNE\FY
Sub-directories:  None
Path: \INNE\SKEY
Sub-directories:  None
Path: \SYS
Sub-directories:  WDWS
                  DOS_3_30
Path: \SYS\WDWS
Sub-directories:  None
Path: \SYS\DOS_3_30
Sub-directories:  None
Path: \SYSUT
Sub-directories:  GUPOTY
                  NORTONY
                  KOPIKI
                  SZYFR
Path: \SYSUT\GUPOTY
Sub-directories:  None
Path: \SYSUT\NORTONY
Sub-directories:  None
Path: \SYSUT\KOPIKI
Sub-directories:  None
Path: \SYSUT\SZYFR
Sub-directories:  None
Path: \TP
Sub-directories:  TPGRAF
                  TLBX
Path: \TP\TPGRAF
Sub-directories:  None
Path: \TP\TLBX
Sub-directories:  None
```

W przypadku istnienia aż tak rozbudowanych katalogów na dyskach ścieżki dostępu mogą bardzo wydłużać się. Spróbujmy zanalizować następujące polecenie:

```
C>a:\sys\xcopy b:\teksty\opisy\*.chi c:\chi\chitxt
```

System operacyjny ma znaleźć polecenie **XCOPY** na dyskietce w napędzie A: i wykonać je. Zadaniem **XCOPY** jest skopiowanie z podkatalogu OPISY katalogu \TEKSTY na dysku B: wszystkich zbiorów o rozszerzeniu nazwy CHI do podkatalogu CHITXT katalogu \CHI na dysku C:. Wiem, że brzmi to strasznie - ale nie ma czym przerażać się - dla ułatwienia wymyślono polecenie **PATH** (wewnętrzne), które wskazuje DOSowi ścieżki dostępu. W naszym przypadku (jeżeli nasza dyskietka z katalogiem \SYS znajduje się w napędzie A:) będzie to

```
A>path a:\sys <ENTER>
```

Tak sformułowane polecenie mówi systemowi operacyjnemu że jeżeli nie zostanie znaleziony żaden wykonywalny (o rozszerzeniu COM, EXE lub BAT) program w aktualnym katalogu aktualnego napędu, to polecenia o podanej nazwie należy szukać w katalogu \SYS na dyskietce w napędzie A:. W jednym poleceniu **PATH** można jednocześnie podać kilka ścieżek dostępu, używając średnika jako separatora między nimi, np.

```
A>path a:\sys;b:\;c:\ados
```

Nasze poprzednie polecenie można zatem nieco uprościć:

```
C>xcopy b:\teksty\opisy\*.chi c:\chi\chitxt
```

Jeszcze mniej pisania będziemy mieli, jeżeli katalogiem aktualnym uczynimy katalog OPISY w napędzie B: (najpierw trzeba przejść na B:, a potem podać polecenie CD\TEKSTY\OPISY, można też od razu podać polecenie CD B:\TEKSTY\OPISY) - całe polecenie będzie miało wtedy postać

```
B>xcopy *.chi c:\chi\chitxt
```

Jeżeli przed zmianą napędu z C: na B: podalibyśmy polecenie CD\CHI\CHITXT, to umożliwiłoby to dalsze skrócenie naszego (na początku tak rozbudowanego) polecenia:

```
B>xcopy *.chi c:
```

Przypominam jednak, że powinny być spełnione następujące warunki:
- jest podana ścieżka dostępu do **XCOPY**,
- na dysku C: jest otwarty katalog \CHI\CHITXT,
- na dyskietce w napędzie B: jest aktualny katalog \TEKSTY \OPISY,
- napędem aktualnym jest napęd B:.

Czytelnik sam zechce zastanowić się, jaki będą skutki wykonania polecenia

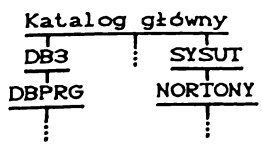
```
C>xcopy b:*.chi
```

przy:
- podanej ścieżce dostępu do **XCOPY**,
- katalogu aktualnym \CHI\CHITXT na C:,
- katalogu otwartym \TEKSTY\OPISY na B:.

Na zakończenie pobieżnego opisu operacji z katalogami dodajmy, że polecenie CD umożliwia zmianę otwartego katalogu na napędzie innym niż aktualny (patrz poprzednia strona), np.

```
A>cd b:\sys <ENTER>
```

oraz zmianę katalogu aktualnego bez przechodzenia przez katalog wspólny dla nowego i starego katalogu aktualnego. Na przykład, jeżeli katalogiem aktualnym jest \DB3\DBPRG przy następującej strukturze drzewa katalogów:



to polecenie

```
C>cd\sysut\nortony <ENTER>
```

od razu powoduje przejście do katalogu \SYSUT\NORTONY (katalogiem wspólnym jest w tym przypadku katalog główny).

Pracę z rozbudowanymi drzewami katalogów możemy sobie ułatwić stosując wewnętrzne polecenie **PROMPT**. Zmienia ono znak zgłoszenia systemu zgodnie z życzeniem użytkownika. Podajmy zatem polecenie

```
A>prompt $p$q$g <ENTER>
```

System odpowie zgłoszeniem

```
A:\=>
```

Takie zgłoszenie systemu daje nam informację nie tylko o aktualnym napędzie, ale i o aktualnym katalogu. Jeżeli pracujemy z dyskietką systemową z utworzonym na niej katalogiem \SYS, to po podaniu polecenia

```
A:\=>cd\sys <ENTER>
```

zgłoszenie systemu przybierze następującą postać:

```
A:\SYS=>
```

Powrót do konwencjonalnego znaku zgłoszenia następuje po podaniu polecenia **PROMPT** bez żadnych parametrów. Podanie polecenia **PROMPT** z parametrem np. WITAM spowoduje, że system będzie zgłaszać się słowem WITAM. My użyliśmy udogodnienia przewidzianego przez twórców systemu - parametr **$** wraz z dodatkowymi znakami umożliwia umieszczenie w zgłoszeniu systemu różnych dodatkowych informacji. Szczegółową listę parametrów polecenia **PROMPT** można znaleźć w rozdziale 3.8 ("Polecenia DOSu").

3.6 Edycja poleceń DOSu - klawisze edycyjne i sterujące

Poznaliśmy już kilka podstawowych poleceń systemu operacyjnego i potrafimy całkiem swobodnie posługiwać się katalogami. Każda operacja, którą chcemy zlecić systemowi operacyjnemu, wymaga pewnego nakładu pracy związanego z posługiwaniem się klawiaturą. Dla większości osób stanowi to niewątpliwie - mniejszy lub większy - problem (oswojenie się z maszyną do pisania może być pomocne). Z tego chyba powodu w Stanach Zjednoczonych niemal każdy kurs obsługi komputerów dla początkujących zaczyna się od nauki sprawnego posługiwania się klawiaturą. Każdemu czytelnikowi, który po klawiaturze stuka jednym lub dwoma palcami, zaleciłbym nieco ćwiczeń - czas temu poświęcony zwróci się potem wielokrotnie. Prostym sposobem umożliwiającym "otrzaskanie się" z klawiaturą jest przepisywanie tekstów. Nie jest do tego konieczne dysponowanie żadnym programem typu word processor (przetwarzania tekstów) - pisany tekst można zapisać na dyskietkę i potem odtworzyć go, korzystając z funkcji DOSu. Spróbujmy użyć ponownie polecenia **COPY**:

```
A:\=>copy con mojzbior.txt <ENTER>
```

Po podaniu tego polecenia z ekranu monitora zniknie znak zgłoszenia systemu i od tej chwili wszystko, co napiszemy na klawiaturze, zostanie umieszczone w tzw. buforze (części pamięci operacyjnej). Możemy teraz - dla uzyskania większej sprawności w posługiwaniu się klawiaturą - przepisywać jakiś tekst (kończąc każdy wiersz wciśnięciem klawisza ENTER). Jeżeli odechce nam się tych ćwiczeń, to wystarczy na zakończenie ostatniego wiersza przepisywanego tekstu również wcisnąć ENTER. Gdy kursor znajdzie się w nowej, pustej linii, wciskamy klawisz funkcyjny F6, dający kod oznaczający koniec zbioru (jest to Ctrl-Z, na ekranie wygląda jak ^Z), a potem ponownie klawisz ENTER. Spowoduje to zapisanie wpisanego przez nas tekstu jako zbioru na dyskietce (zbiór będzie nosić nazwę MOJZBIOR.TXT, zgodnie z tym, co określiliśmy w poleceniu COPY). Plik ten możemy obejrzeć na ekranie (w celu wykrycia popełnionych błędów), podając polecenie

```
A:\=>type mojzbior.txt <ENTER>
```

Jeżeli ćwiczyliśmy solidnie i wpisany przez nas tekst nie mieści się na ekranie, to wyświetlanie możemy zatrzymywać, wciskając <Ctrl-S> (klawisz Ctrl i jednocześnie z nim klawisz oznaczony literą S). Ten sam efekt uzyskujemy, wciskając <Ctrl-NumLock>. Wyświetlanie tekstu będzie kontynuowane po wciśnięciu jakiegokolwiek klawisza (np. spacji). Jeżeli chcemy wydrukować rezultaty naszych ćwiczeń z klawiaturą (i posiadamy drukarkę), możemy podać następujące polecenie:

```
A:\=>copy mojzbior.txt prn <ENTER>
```

Najlepsza nawet maszynistka może pomylić się przy wypisywaniu nazwy polecenia. Dlatego system operacyjny przewiduje możliwość przeprowadzania edycji podawanych poleceń i zawiera w sobie mechanizmy, które ułatwiają intensywną pracę z nim.

Do usuwania niewłaściwych znaków z wiersza poleceń służy klawisz Backspace, opisany na początku rozdziału.

Klawisz funkcyjny F1 umożliwia odtwarzanie ostatnio podanego polecenia znak po znaku. Przez kilkukrotne wciśnięcie F1 możemy odtworzyć ostatnie polecenie aż do tego miejsca, które chcemy zmienić.

Wciśnięcie klawisza funkcyjnego F3 powoduje ponowne wyświetlenie na ekranie treści ostatniego polecenia w całości (treść polecenia jest tylko wyświetlana - samo polecenie nie jest wykonywane), niezależnie od tego, czy polecenie było prawidłowe, czy nie. Jeżeli w linii poleceń (za znakiem zgłoszenia się systemu) jest wpisywane nowe polecenie, to wciśnięcie klawisza F3 spowoduje odtworzenie tylko tylu ostatnich znaków uprzednio podanego polecenia, o ile było ono dłuższe od polecenia aktualnie wpisywanego.

Wciśnięcie klawisza funkcyjnego F2 i po nim znaku z klawiatury powoduje odtworzenie ostatniego polecenia aż do miejsca wystąpienia tegoż znaku (może to być litera, dwukropek, średnik itd.). Wykonanie takiej samej czynności z klawiszem funkcyjnym F4 powoduje usunięcie tekstu ostatniego polecenia aż do miejsca napotkania znaku podanego po wciśnięciu klawisza funkcyjnego.

Klawisz F5 wprowadza do bufora polecenie wpisane przez użyt-

kownika, a nie zakończone wciśnięciem klawisza ENTER (polecenie takie nie jest wykonywane).

Jednoczesne wciśnięcie klawiszy Ctrl i S (lub Ctrl i NumLock) powoduje wstrzymanie aktualnie wykonywanego polecenia - np. wstrzymanie wyświetlania zawartości jakiegoś zbioru.

Jednoczesne wciśnięcie klawiszy Ctrl i C (lub Ctrl i Scroll Lock), oznaczane też jako <Ctrl-Break>, powoduje przerwanie wykonywania polecenia.

Wciśnięcie klawisza Esc (ang. escape) podczas wpisywania polecenia powoduje unieważnienie napisanego tekstu i przejście do nowego wiersza, gdzie wprowadzanie polecenia można rozpocząć ponownie. Wciśnięcie ENTER po Esc spowoduje ponowne pojawienie się znaku zgłoszenia się systemu. Nie można wprowadzać poleceń o długości większej, niż 127 znaków.

Podczas przeprowadzania edycji poleceń DOSu można korzystać z klawiszy Ins (ang. insert, czyli wstaw) i Del (ang. delete, czyli usuń).

Teraz przejdźmy do przykładów. Podajmy polecenie

```
A:\=>dir *.txt <ENTER>
```

Zostanie wyświetlony wykaz wszystkich zbiorów o rozszerzeniu nazwy TXT. Jeżeli chcielibyśmy skasować te zbiory, to możemy napisać

```
A:\=>del
```

i wcisnąć klawisz funkcyjny F3. Po wciśnięciu tego klawisza do polecenia DEL zostanie dopisane *.TXT:

```
A:\=>del *.txt
```

W celu unieważnienia tego polecenia nie musimy kasować (za pomocą klawisza Backspace) po kolei wszystkich znaków w nim zawartych - wystarczy raz wcisnąć klawisz Esc i linia poleceń będzie wyglądać następująco:

```
A:\=>del *.txt\
```

Zaczerniony prostokąt przedstawia kursor, który został przeniesiony jedną linię niżej. System jest gotowy do przyjęcia nowego polecenia, a DEL *.TXT zostało już "zapomniane". Tzn. można już wpisywać nowe polecenie, ale DEL *.TXT nadal można przywołać przez wciśnięcie F3. Wciśnięcie ENTER w pustym wierszu spowoduje usunięcie dotychczasowej zawartości bufora (tym razem DEL *.TXT zostanie "zapomniane" na dobre) i ponowne pojawienie się znaku zgłoszenia się systemu.

Jeżeli zbiór MOJZBR.TXT jest zbiorem tekstowym, to możemy napisać:

```
A:\=>type mojjzbr.txt <ENTER>
```

Na takie polecenie system odpowie komunikatem

```
File not found
```

czyli

```
Zbiór nie został znaleziony
```

Podczas wpisywania nazwy zbioru popełniony został błąd i system nie znalazł żadnego zbioru o podanej nazwie, do którego mógłby zastosować polecenie TYPE. Spróbujmy poprawić naszą pomyłkę - wciśnijmy kolejno klawisze

```
F2
j   ] odtworzenie polecenia aż do wystąpienia litery j
DEL   usunięcie drugiego j
F3    odtworzenie polecenia do końca
```

Skutek powinien być taki:

```
A:\=>type mojzbr.txt
```

Prześledźmy inny przykład. Przypuśćmy, że z dyskietki w napędzie A: chcemy skopiować zbiór NOWY.TXT jako STARY.TXT na dyskietkę w napędzie B:. Podczas podawania polecenia pomyliliśmy się:

```
A:\=>copy stary.txt b:nowy.txt <ENTER>
```

System odpowie komunikatem o błędzie (na A: nie ma zbioru STARY.TXT). Polecenie powinno mieć postać

```
A:\=>copy nowy.txt b:stary.txt
```

W celu doprowadzenia polecenia do takiej postaci wciskamy kolejno

```
F2   s   F4   n   F3
```

i dopisujemy

```
b:stary.txt
```

W ten sposób powinniśmy uzyskać żądaną przez nas postać polecenia.
Zróbmy ostatni przykład - załóżmy, że podaliśmy polecenie

```
A:\=>\ss\format b//v\s <ENTER>
```

Naszym zamiarem jest sformatowanie dyskietki w napędzie B: i przeniesienie na nią systemu. Nie chcemy nadawać dyskietce nazwy. Zbiór FORMAT.COM znajduje się w katalogu \SYS (a nie \SS) i konieczne jest podanie systemowi ścieżki dostępu do tego zbioru. Po wciśnięciu ENTER system wyprowadzi komunikat o błędzie. W związku z tym nasze polecenie powinno być sformułowane następująco:

```
A:\=>\sys\format b:/s
```

Żeby doprowadzić je do takiej postaci, należy kolejno wcisnąć:

```
F1   F1   Ins   y   F2   /       F4   s   Ins   /   F1
```

Czytelnik sam zechce zastanowić się, w jaki sposób można szybciej i efektywniej uzyskać żądaną postać polecenia (podpowiem tylko, że konieczne będzie użycie klawisza DEL).

Trzeba przyznać, że używanie klawiszy edycyjnych DOSu nie zawsze może wydawać się celowe, ale - jak wynika z praktyki -

najczęściej występujące pomyłki to "o jeden znak za dużo", "o jeden znak za mało" lub "jeden znak niewłaściwy". W takich przypadkach użycie klawiszy edycyjnych jest bardzo wygodne. Pierwszy przykład podany w tej części rozdziału 3 (DIR *.TXT i DEL *.TXT) świadczy o zwiększeniu pewności pracy użytkownika komputera dzięki zastosowaniu klawiszy edycyjnych - przed skasowaniem określonej grupy zbiorów można zawsze najpierw zobaczyć nazwy poszczególnych zbiorów wchodzących w skład tej grupy (polecenie DIR *.TXT). Jeżeli po napisaniu polecenia DEL użyjemy klawisza F3, to mamy pewność, że zostaną usunięte tylko te zbiory, których nazwy widzieliśmy uprzednio. Nie ma niebezpieczeństwa popełnienia pomyłki przy ponownym określaniu wzorca nazw zbiorów mających ulec kasowaniu (po podaniu polecenia DIR *.TXT można omyłkowo napisać np. DEL *.*, co spowodowałoby skasowanie wszystkich zbiorów na dyskietce - użycie F3 chroni nas przed popełnieniem takiego błędu).

3.7 Potoki i filtry w DOSie

W DOSie używane są tzw. filtry i potoki (zwane niekiedy strumieniami).

Filtry pobierają dane z urządzeń wejścia (np. zbiór z dysku), modyfikują je zgodnie z określonym algorytmem działania (np. przeprowadzają sortowanie, wyszukują pewne elementy zbioru) i wysyłają wynik swojego działanie na urządzenie wyjścia (np. do innego zbioru na dysku, na drukarkę, na monitor).

Potoki umożliwiają przełączanie strumieni danych przepływających między urządzeniami wejścia/wyjścia,programami, poleceniami itd.

Do przeprowadzania operacji z filtrami i potokami są stosowane pewne znaki przeznaczone do tych operacji. Są to >, >>, < i |. Znaki >, >> i < są czymś w rodzaju rur, którymi płyną dane. Taką rurą można wyjście (wynik wykonania) polecenia skierować na dowolne urządzenie wyjścia lub do zbioru. Z rury do polecenia (np. filtru) mogą wpłynąć dane, na których polecenie ma pracować. Istnieje też możliwość podłączenia wyjścia jednego polecenia do wejścia innego polecenia (za pomocą |).

Poznajmy kilka prostych przykładów. Jeżeli chcemy, żeby katalog dyskietki znalazł się w zbiorze SPIS.KAT, to możemy podać polecenie

```
A:\=>dir > spis.kat <ENTER>
```

Na ekranie nie zobaczymy właściwie żadnych efektów działania polecenia DIR - wyjście polecenia zostało skierowane do zbioru SPIS.KAT przez >. Zbiór ten zostanie utworzony, jeżeli nie istniał wcześniej. Użycie polecenia

```
A:\=>dir b: > spis.kat <ENTER>
```

spowoduje zniszczenie poprzedniej zawartości SPIS.KAT i wpisanie na jej miejsce nowej. Jeżeli nie chcemy tracić tego, co było zapisane w SPIS.KAT do tej pory (np. katalog dyskietki w A:), to należy użyć znaku >> w sposób następujący:

```
A:\=>dir b: >> spis.kat <ENTER>
```

Dzięki temu katalog dyskietki ze stacji B: zostanie dopisany do zbioru SPIS.KAT. Strumień można podłączyć także do urządzenia zewnętrznego, np. wyprowadzenie katalogu na drukarkę powoduje polecenie

```
A:\=>dir > prn <ENTER>
```

a wyprowadzenie zawartości zbioru SPIS.KAT na drukarkę polecenie

```
A:\=>type spis.kat > prn <ENTER>
```

Znak | steruje przepływem danych między poleceniami. Polecenie

```
A:\=>dir | sort <ENTER>
```

powoduje podłączenie wyjścia polecenia DIR do wejścia polecenia SORT i wyprowadzenie alfabetycznie uporządkowanego katalogu na ekran. Posortowany katalog możemy zapisać w zbiorze:

```
A:\=>dir | sort > spis.srt <ENTER>
```

Możliwe jest też sortowanie zawartości zbioru:

```
A:\=>sort < spis.kat > spis.srt <ENTER>
```

Takie polecenie zapisze posortowaną zawartość zbioru SPIS.KAT w zbiorze SPIS.SRT.

3.8 Polecenia DOSu

Do tej pory poznaliśmy już kilka podstawowych poleceń DOSu wraz z ich zastosowaniem. Nie mamy jednak usystematyzowanej wiedzy o ogólnych zasadach, zgodnie z którymi funkcjonuje całe otoczenie polecenia (ścieżki dostępu, strumienie, filtry itd.), nie znamy jeszcze wszystkich poleceń z ich wszystkimi możliwościami. W tej części rozdziału 3 postaram się przedstawić listę poleceń DOSu wraz z krótkim omówieniem ich składni i zastosowania. Najpierw jednak nieco wiadomości wstępnych.

Zacznijmy od oznaczeń, które będą stosowane w dalszej części książki:

- nawiasy kwadratowe [ ] - napis w nawiasach kwadratowych oznacza opcjonalny (wprowadzany tylko w razie potrzeby) element polecenia,
- **DUŻE LITERY** (pogrubione) - słowa pisane pogrubionymi dużymi literami są słowami kluczowymi (nazwami poleceń) DOSu,
- *kursywa* - napis kursywą symbolizuje element polecenia, który użytkownik powinien uzupełnić (np. *nzwzbru.roz* oznacza nazwę zbioru wraz z rozszerzeniem),
- pionowa kreska | (tzw. belka) - elementy polecenia oddzielone pionową kreską (oznacza ona "albo") zastępują się nawzajem, np. napis 8|9 mówi o konieczności wpisania albo cyfry 8, albo cyfry 9,
- wielokropek ... - oznacza możliwość wielokrotnego powtórzenia elementu polecenia,
- d: - jest oznaczeniem stacji dysków,
- path - oznacza ścieżkę dostępu.

Wykonanie polecenia DOSu kończy się pomyślnie, jeżeli przed ponownym pojawieniem się znaku zgłoszenia się systemu nie został wyprowadzony żaden komunikat o wystąpieniu błędu.

Wszystkie polecenia operujące na zbiorach (grupach zbiorów) o wskazywanej nazwie pozwalają na umieszczenie ścieżki dostępu do zbioru przed nazwą.

Polecenia i parametry poleceń muszą być rozdzielone separatorami (spacją, przecinkiem, średnikiem, znakiem równości). Ze względu na czytelność zaleca się używanie znaku spacji.

Żadne polecenie nie jest wykonywane aż do momentu wciśnięcia klawisza ENTER.

<Ctrl-S> i <Ctrl-C> (<Ctrl-NumLock> i <Ctrl-ScrollLock>) podczas wykonywania polecenia działają tylko wtedy, gdy system wykonuje standardowe operacje we/wy. Dlatego w celu umożliwienia przerywania wykonywania poleceń DOSu w każdej niemal chwili, należy podać polecenie **BREAK ON**.

Przez "napęd źródłowy" jest rozumiana dyskietka włożona do napędu czytającego z niej dane. "Napęd docelowy" oznacza dyskietkę włożoną do napędu zapisującego na niej dane.

Polecenia DOSu dzielą się (jak już mówiliśmy) na wewnętrzne i zewnętrzne. Polecenia wewnętrzne są realizowane przez procesor poleceń COMMAND.COM, obecny w pamięci operacyjnej komputera. Polecenia zewnętrzne są realizowane przez zbiory programowe o odpowiednich nazwach (i o rozszerzeniu nazwy COM, EXE lub BAT). Zbiór taki przed wykonaniem programu w nim zawartego musi zostać najpierw wczytany z dysku do pamięci. Podając polecenie zewnętrzne nie trzeba podawać rozszerzenia nazwy.

W dalszej części tego rozdziału zostaną opisane polecenia PC DOS 3.10.

ASSIGN

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**ASSIGN** [x[=[y[...]]

Funkcja: polecenie służy do przyporządkowania odwołań programowych do jednej stacji dysków innej stacji. Jeżeli podamy polecenie

```
A:\=>assign a=c b=c
```

to wszelkie odwołania we/wy do jakichkolwiek zbiorów na napędach A: i B: będą realizowane na napędzie C:. Polecenie **DIR A:** spowoduje wyświetlenie listy zbiorów zapisanych na dysku C:. Powrót do stanu wyjściowego uzyskujemy przez podanie polecenia **ASSIGN** bez żadnych parametrów. Użycie poleceń **BACKUP, JOIN, LABEL, SUBST, RESTORE** i **PRINT** po zmianie przyporządkowania odwołań może sprawić pewne kłopoty. Z tego powodu polecenia **ASSIGN** należy używać tylko wtedy, gdy jest to konieczne. Nie zaleca się używania **ASSIGN** podczas normalnej pracy z systemem operacyjnym.

ATTRIB

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**ATTRIB** [+r|-r][*d:*][*path*]*nazwabru*[.*roz*]

Funkcja: zmiana atrybutów zbioru (grupy zbiorów). Podanie parametru +r ustawia atrybut wyspecyfikowanego zbioru na "read-only" (oznacza to "zbiór tylko do odczytu"). Zbiór jest wtedy chroniony przed zniszczeniem danych w nim zawartych przez skasowanie go lub zapisanie w nim innych danych. Atrybut "read-only" uniemożliwia kasowanie zbioru oraz wykonywanie na nim operacji zapisywania. Parametr -r ustawia atrybut wyspecyfikowanego zbioru ponownie na "read-write" (zapis i odczyt). Podanie polecenia

```
A:\=>attrib +r c:\mojzbr.txt <ENTER>
```

ustawia atrybut zbioru MOJZBR.TXT na dysku C: na "read-only". Podanie polecenia

```
A:\=>attrib c:\mojzbr.txt <ENTER>
```

da wtedy następujacy rezultat:

```
R          C:\MOJZBR.TXT
```

Litera R oznacza, że zbiór MOJZBR.TXT ma atrybut "read-only".

Jeżeli zmienimy atrybut tego zbioru ponownie na "read-write" przez podanie polecenia

```
A:\=>attrib -r c:\mojzbr.txt <ENTER>
```

to odpowiedzą na polecenie

```
A:\=>attrib c:\mojzbr.txt <ENTER>
```

będzie komunikat

```
        C:\MOJZBR.TXT
```

oznaczający, że zbiór MOJZBR.TXT ma atrybut "read-write".

BACKUP

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**BACKUP** *d:*[*path*][*nzwzbru*[*.roz*]] *d:*[/s] [/m][/a][/d:mm-dd-rr]

Funkcja: tworzenie rezerwowych kopii zbiorów z jednego dysku na innym. **BACKUP** działa przy tworzeniu kopii zbiorów z

- twardego dysku na dyskietki
- twardego dysku na twardy dysk
- dyskietki na twardy dysk
- dyskietki na dyskietkę

Kopie zbiorów utworzone poleceniem **BACKUP** nie są identyczne z oryginałami i nie mogą być normalnie używane (zawierają pewne dodatkowe informacje będące wynikiem kontroli np. poprawności przeprowadzonego procesu kopiowania). W celu przywrócenia kopii do stanu używalności należy zastosować polecenie **RESTORE**, omówione w dalszej części niniejszego rozdziału.
W parametrach polecenia **BACKUP** zawsze należy podać nazwy dwóch różnych napędów dysków. Pierwsza nazwa napędu za poleceniem **BACKUP** jest nazwą dysku źródłowego, nazwa druga jest nazwą dysku docelowego. Użycie parametru /s powoduje utworzenie kopii także tych zbiorów, które znajdują się w podkatalogach katalogu otwartego na napędzie źródłowym.
Użycie parametru /m powoduje uwzględnienie w procesie tworze-

nia kopii rezerwowych tylko tych zbiorów, które były modyfikowane po ostatnim kopiowaniu ich poleceniem **BACKUP**.
Użycie parametru /a powoduje dodanie tworzonych kopii zbiorów z dysku źródłowego do zbiorów już istniejących na dysku docelowym (jeżeli nie podamy parametru /a, to **BACKUP** spowoduje skasowanie wszystkich zbiorów uprzednio istniejących na dysku docelowym).
Użycie parametru /d umożliwia utworzenie kopii rezerwowych tylko tych zbiorów, które były modyfikowane po określonej dacie.
Po zakończeniu działania polecenia **BACKUP** parametr systemowy ERRORLEVEL jest ustawiany na:

0 - jeżeli kopiowanie przebiegło poprawnie,
1 - jeżeli nie zostały znalezione żadne zbiory, z których można by było utworzyć kopie,
2 - jeżeli niektóre zbiory nie zostały skopiowane (np. ze względu na konflikt podziału podczas pracy komputera w sieci),
3 - jeżeli wykonywanie polecenia zostało przerwane przez użytkownika (<Ctrl-C>),
4 - jeżeli wykonanie polecenia nie było możliwe.

Parametr ERRORLEVEL (dosłownie "poziom błędu") jest przewidziany do wykorzystania głównie w zbiorach przetwarzania wsadowego (zbioy typu BAT).
BACKUP umożliwia efektywne wykorzystanie dyskietek, ponieważ dyskietki docelowe (w przypadku kopiowania dużej ilości zbiorów) są zapełniane całkowicie.
Używając polecenia **BACKUP** należy pamiętać o tym, że:
- wywołanie polecenia bez parametru /a spowoduje skasowanie wszystkich zbiorów znalezionych na dysku docelowym,
- w przypadku wykonywania kopii dużej ilości zbiorów kolejne dyskietki docelowe należy ponumerować (**BACKUP** żąda włożenia następnej dyskietki, jeżeli zabraknie miejsca na kopie), ponieważ polecenie **RESTORE** przywracające kopie rezerwowe do stanu normalnego wymaga zachowania kolejności odtwarzania zgodnej z kolejnością tworzenia kopii rezerwowych,
- przy używaniu miękkigo dysku (dyskietki) jako dysku źródło-

wego nie można zabezpieczać go przed zapisaniem (BACKUP aktualizuje pewne dodatkowe informacje, używane np. w przypadku wywołania tego polecenia z parametrem /m),
- gdy dyskiem docelowym jest twardy dysk, kopie rezerwowe są umieszczane w katalogu \BACKUP.
- nie należy używać polecenia **BACKUP**, jeżeli są aktywne polecenia **JOIN** lub **ASSIGN**.

BREAK

Polecenie wewnętrzne.

Składnia: **BREAK** [**ON** | **OFF**]

Funkcja: polecenie ustawia tryb kontroli pojawienia się przerwania użytkownika <Ctrl-C> (<Ctrl-Break>). Standardowo **BREAK** jest ustawiane na **OFF** - wtedy system operacyjny sprawdzą, czy z klawiatury nie przychodzi sygnał <Ctrl-C> tylko podczas
- standardowych operacji wejścia/wyjścia,
- standardowej obsługi portu drukarki,
- standardowej obsługi portu komunikacji szeregowej.

Ustawienie **BREAK** na **ON** sprawia, że jest możliwe przerwanie wykonywania dowolnego polecenia niemal w każdej chwili. Polecenie **BREAK** podane bez żadnych parametrów powoduje wyświetlenie informacji o aktualnym ustawieniu trybu kontroli.

CHDIR lub CD

Polecenie wewnętrzne.

Składnia: **CHDIR** [*d:*][*path*]
lub
CD [*d:*][*path*]

Funkcja: zmiana katalogu aktualnego w napędzie aktualnym lub zmiana katalogu otwartego w każdym innym napędzie. Podanie polecenia

A:\=>cd

powoduje wyświetlenie ścieżki dostępu do katalogu aktualnego. Podanie polecenia

```
A:\=>cd\ <ENTER>
```

powoduje przejście do katalogu głównego w napędzie aktualnym. Polecenie

```
A:\SYSUT\GUPOTY=>cd\chi\chitxt <ENTER>
```

powoduje zmianę katalogu aktualnego z \SYSUT\GUPOTY na \CHI\CHITXT. Otworzenie katalogu \SYS w napędzie B: (zakładamy, że A: jest napędem aktualnym) możemy uzyskać, podając polecenie

```
A:\=>cd b:\sys <ENTER>
```

Jeżeli przed użyciem CD korzystaliśmy z poleceń **ASSIGN**, **JOIN** i **SUBST**, mogą wystąpić nieprawidłowości w działaniu CD.

CHKDSK

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]CHKDSK [*d:*][*path*][*nazwapliku*[.*roz*]][/f] [/v]

Funkcja: przeprowadzenie analizy katalogów, zbiorów i obszaru FAT (File Allocation Table, czyli tablica rozmieszczenia zbiorów) w napędzie aktualnym lub podanym, po czym zostaje wyświetlony raport o stanie dysku i pamięci operacyjnej. Jeżeli jako parametr zostanie podana tylko nazwa zbioru, to **CHKDSK** wyświetla liczbę nieciągłych obszarów zajmowanych przez zbiór (informacja o fragmentacji zbioru).
Użycie **CHKDSK** z parametrem /f powoduje skorygowanie błędów znalezionych na dysku.
Użycie parametru /v powoduje wyświetlenie nazw zbiorów i ścieżek dostępu do nich w aktualnym lub podanym napędzie.

CLS

Polecenie wewnętrzne.

Składnia: **CLS**

Funkcja: czyszczenie ekranu.

COMMAND

Polecenie zewnętrzne.

Składnia: **COMMAND** [*d:*][*path*][*nazwa*][/p][/c *łańcuch znaków*]

Funkcja: zmienia procesor poleceń ze standardowego COMMAND.COM na inny o podanej *nazwie*.

Użycie parametru /p uniemożliwia powrót do COMMAND.COM spod nowego procesora poleceń.

Użycie parametru /c powoduje przesłanie łańcucha znaków (np. polecenia) do innego procesora poleceń. Wykonuje on przesłane mu polecenie i powraca do standardowego procesora poleceń.

Użycie parametru /c razem z /p powoduje unieważnienie tego ostatniego.

Polecenie **COMMAND** jest używane bardzo rzadko i ma znaczenie jedynie dla osób tworzących oprogramowanie systemowe oraz wyrafinowane w działaniu zbiory przetwarzania wsadowego.

Dokładne informacje o poleceniu **COMMAND** można znaleźć w *DOS Technical Reference*, rozdział 7.

COMP

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**COMP** [*d:*][*path*][*nzwzbru*[.*roz*]][*d:*]
[*path*][*nzwzbru*[.*roz*]]

Funkcja: porównanie zawartości dwóch wyszczególnionych zbiorów. Jest również możliwe porównywanie zawartości kolejnych zbiorów pierwszej i drugiej wyspecyfikowanej grupy zbiorów. W przypadku identycznej zawartości porównywanych zbiorów, jest wyświetlany komunikat FILES COMPARE OK. Wystąpienie niezgodności powoduje wyprowadzenie komunikatów informujących o różnicach zawartości porównywanych zbiorów. Porównywanie zbiorów zostaje przerwane, gdy zostanie znalezionych 10 różniących się bajtów.

COPY

Polecenie wewnętrzne.

Składnia: **COPY** [/a][/b][*d:*][*path*]*nzwzbru*.[*roz*][/a][/b]
[*d:*][*path*][*nzwzbru*.[*roz*]][/a][/b][/v]

lub

COPY [/a][/b][*d:*][*path*]*nzwzbru*.[*roz*][/a][/b]

[+[*d:*][*path*]*nzwzbru.*[*roz*][/a][/b]...]
[*d:*][*path*][*nzwzbru.*[*roz*]][/a][/b][/v]

lub

COPY [/a][/b][*d:*][*path*]*nzwzbru.*[*roz*][/a][/b]
[+[[,,]*d:*][*path*]*nzwzbru.*[*roz*][/a][/b]
...] [*d:*][*path*][*nzwzbru.*[*roz*]][/a][/b][/v]

Funkcja: kopiowanie zbiorów i przesyłanie danych między urządzeniami we/wy.
Istnieje możliwość konkatenacji (łączenia) zbiorów za pomocą znaku +.
Użycie parametru /v powoduje prowadzenie weryfikacji przesyłanych danych (działanie takie samo, jak polecenia VERIFY ON).
Użycie parametrów /a i /b powoduje traktowanie kopiowanych zbiorów odpowiednio jako zbiorów typu znakowego i binarnego (ang. binary).

CTTY

Polecenie wewnętrzne.

Składnia: **CTTY** *nazwa-urządzenia*

Funkcja: zmienia standardowe urządzenie we/wy (CON: - konsola) na urządzenie wyspecyfikowane. Urządzenie to musi być zdolne do przeprowadzania operacji wejścia i wyjścia (zatem nie może to być np. drukarka). Polecenie CTTY działa tylko z programami używającymi wywołań procedur obsługi urządzeń we/wy używanych przez DOS. Podanie polecenia

A:\=>ctty com1 <ENTER>

spowoduje, że standardowym urządzeniem we/wy stanie się urządzenie podłączone do portu COM1:. Polecenie

A:\=>ctty con <ENTER>

ponownie przypisze wykonywanie standardowych operacji we/wy konsoli.

DATE

Polecenie wewnętrzne.

Składnia: **DATE** [*mm-dd-rr*]|[*dd-mm-rr*]|[*rr-mm-dd*]

Funkcja: wprowadzanie lub zmiana daty używanej przez DOS. Format daty jest uzależniony od polecenia **SELECT**, które dostosowuje układ klawiatury i format daty do ustaleń obowiązujących w określonych krajach, lub od poleceia **COUNTRY**.

DEL

Polecenie wewnętrzne.

Składnia: **DEL** [*d:*][*path*]*nzwzbru.*[*roz*]

Funkcja: kasowanie wyspecyfikowanego zbioru (grupy zbiorów). System nie pyta o potwierdzenie polecenia usunięcia określonego zbioru (zbiorów). Wyjątek stanowi wydanie polecenia usunięcia wszystkich zbiorów w katalogu:

A:\=>del *.* <ENTER>

W takim przypadku system odpowie komunikatem

Are you sure (Y/N)?

czyli

Jesteś pewien (T/N)?

Jeżeli rzeczywiście chcemy usunąć wszystkie zbiory w aktualnym katalogu, wciskamy klawisz y i ENTER. Nie można usunąć zbiorów oznaczonych jako "read-only". Polecenie **DEL** powinno b używana szczególnie ostrożnie po zastosowaniu poleceń **ASSIGN**, **JOIN** i **SUBST**.

DIR

Polecenie wewnętrzne.

Składnia: **DIR** [*d:*][*path*]*nzwzbru.*[*roz*][/p][/w]

Funkcja: wyświetlanie wszystkich lub wyspecyfikowanych elementów katalogu (zbiorów i podkatalogów).
Użycie parametru /p powoduje wstrzymanie wyświetlania po zapełnieniu całego ekranu (kontynuacja wyświetlania po wciśnięciu dowolnego klawisza).

Użycie parametru /w powoduje wyświetlanie nazw zbiorów i podkatalogów w tzw. szerokim formacie (są wyświetlane tylko nazwy wraz z rozszerzeniami, w pięciu kolumnach, bez żadnych informacji dodatkowych).

DISKCOMP

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**DISKCOMP** [*d:*[*d:*]][/1][/8]

Funkcja: porównywanie zawartości dyskietek ścieżka po ścieżce. Możliwe jest porównanie dyskietek jedno- (parametr /1) i dwustronnych, z 9 i z 8 (parametr /8) sektorami na ścieżkę oraz o pojedynczej, podwójnej i poczwórnej (wysokiej) gęstości. Ponieważ porównywanie jest dokonywane kolejno "ścieżka po ścieżce", to obydwie dyskietki muszą być tego samego typu.
Nie można używać polecenia **DISKCOMP** przy aktywnym poleceniu **JOIN**.

DISKCOPY

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**DISKCOPY** [*d:* [*d:*]][/1]

Funkcja: kopiowanie dyskietek. Kopiowanie może odbywać się przy użyciu tak jednego, jak i dwóch napędów i dotyczyć albo obu stron, albo tylko pierwszej (gdy zostanie użyty parametr /1) strony dyskietki dowolnego typu. Jeżeli dyskietka docelowa różni się formatem od dyskietki źródłowej, to podczas kopiowania zostanie ona sformatowana zgodnie z formatem dyskietki źródłowej.
Polecenia **DISKCOPY** nie należy używać, gdy są aktywne polecenia **ASSIGN**, **SUBST** i **JOIN**.
Nie zaleca się używania **DISKCOPY** do kopiowania dyskietek uprzednio intensywnie wykorzystywanych. Jeżeli na dyskietce odbywało się wielokrotne usuwanie i kreowanie zbiorów, to skutkiem takiej działalności jest ich silna fragmentacja. Następstwem tego jest dłuższy czas dostępu do zbiorów, obniżający sprawność całego systemu komputerowego. Użycie polecenia **DISKCOPY** do dyskietki o znacznej fragmentacji zbiorów

spowoduje przeniesienie tej fragmentacji na dyskietkę docelową. Aby uniknąć fragmentacji, można sformatować nową dyskietkę i zastosować polecenie **COPY** kopiujące zbiory sekwencyjnie (zbiór po zbiorze).
DISKCOPY nie może odnosić się ani do dysku wirtualnego, ani do dysku twardego.

ERASE - jak **DEL**

EXE2BIN

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**EXE2BIN** [*d:*][*path*]*nzwzbru*[*.roz*] [*d:*][*path*][*nzwzbru*[*.roz*]]

Funkcja: konwersja zbioru typu EXE na zbiór COM lub BIN. Pierwszy wyszczególniony zbiór jest zbiorem wejściowym, zaś drugi - wyjściowym.

FDISK

Polecenie omówione w części "Uruchomienie systemu z twardym dyskiem".

FIND

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**FIND** [/v][/c][/n]"*łańcuch znaków*"[[*d:*][*path*]*nzwzbru*[*.roz*]...]

Funkcja: Wyszukanie danego łańcucha znaków w danym zbiorze (danej grupie zbiorów). Polecenie **FIND** należy do grupy poleceń określanych mianem "filtrów" i będących przydatnymi w nieco bardziej wyrafinowanych zastosowaniach systemu operacyjnego.
Użycie parametru /v powoduje wyświetlenie tych wierszy zbioru, które nie zawierają szukanego łańcucha.
Użycie parametru /c powoduje wyprowadzenie informacji o liczbie wierszy zawierających wyspecyfikowany łańcuch znaków. Jeżeli parametr ten zostanie użyty, to inne parametry są ignorowane.

Użycie parametru /n powoduje wyświetlenie wierszy zawierających wyspecyfikowany łańcuch znaków wraz z relatywnymi numerami tych wierszy.
W nazwach zbiorów, które mają być przeszukane, nie można stosować znaków odnoszących się do całych grup zbiorów (? i *).
Polecenie

```
A:\=>find "XLV etap reformy" mowa1.txt mowa2.txt
```

jest poprawne, natomiast polecenie

```
A:\=>find "XLV etap reformy" mowa?.*
```

jest sformułowane błędnie.
Prześledźmy kilka przykładów zastosowania polecenia FIND. Załóżmy, że nasz zbiór tekstowy o nazwie DEK.TXT przedstawia się następująco:

```
{$I typedef.sys}
{$I graphix.sys}
{$I kernel.sys}
{$I windows.sys}
{{$I dummy.inc}
```

Jeżeli podamy polecenia

```
1)   A:\=>find /n"x"dek.txt <ENTER>
2)   A:\=>find /c"x"dek.txt <ENTER>
3)   A:\=>find /v"x"dek.txt <ENTER>
4)   A:\=>find /v"d"dek.txt <ENTER>
5)   A:\=>find /c"d"dek.txt <ENTER>
6)   A:\=>find /n"d"dek.txt <ENTER>
```

to odpowiednio zostaną wyświetlone komunikaty:

```
1)
---------- dek.txt
[2]{$I graphix.sys}

2)
---------- dek.txt: 1
```

```
3)
---------- dek.txt
{$I typedef.sys}
{$I kernel.sys}
{$I windows.sys}
{{$I dummy.inc}

4)
---------- dek.txt
{$I graphix.sys}
{$I kernel.sys}

5)
---------- dek.txt: 3

6)
---------- dek.txt
[1]{$I typedef.sys}
[4]{$I windows.sys}
[5]{{$I dummy.inc}
```

FORMAT

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**FORMAT** [*d:*][/s][/1][/8][/v][/b][/4]

Funkcja: polecenie służy do przygotowania dyskietek (twardego dysku) do używania ich przez DOS. **FORMAT** szuka na dyskietce wadliwych ścieżek, inicjalizuje obszary katalogu, FAT i programu ładującego. Wadliwe ścieżki są oznaczane podczas formatowania, aby później możliwe było zapobieżenie wykorzystaniu ich do zapisu danych. Z dyskietki uprzednio zapisanej i ponownie sformatowanej są tracone wszystkie dane, dlatego polecenie **FORMAT** należy stosować bardzo ostrożnie i z pełną świadomością tego, którą dyskietkę formatujemy. Pamiętajmy o tym, że samo polecenie **FORMAT** (bez wyspecyfikowanego napędu, w którym ma odbyć się formatowanie) przeprowadza formatowanie na napędzie aktualnym. Na tę właściwość polecenia **FORMAT** powinni być szczególnie uczuleni użytkownicy twardych dysków, którzy na każde pytanie wyprowadzone przez system operacyjny odruchowo odpowiadają "Y".

Oznaczenie napędu, w którym chcemy przeprowadzić formatowanie umieszczamy za poleceniem **FORMAT**.
Użycie parametru /s powoduje przeniesienie zbiorów systemowych IBMBIO.COM, IBMDOS.COM i COMMAND.COM w tej właśnie kolejności na formatowaną dyskietkę. W napędzie aktualnym musi znaleźć się dyskietka z rezydującym na niej systemem (może to być system na twardym dysku), aby przeniesienie systemu było możliwe. Jeżeli **FORMAT** nie znajduje zbiorów systemowych w napędzie aktualnym, to jest wyświetlany komunikat informujący użytkownika o konieczności włożenia dyskietki ze zbiorami systemowymi do napędu aktualnego.
Użycie parametru /1 powoduje jednostronne sformatowanie dyskietki.
Użycie parametru /8 powoduje formatowanie dyskietki z ośmioma sektorami na ścieżkę. Standardowo używanych jest 9 (dyskietki o podwójnej gęstości zapisu) i 15 (dyskietki o dużej gęstości zapisu) sektorów na ścieżkę.
Użycie parametru /v umożliwia nadanie nazwy nowo sformatowanej dyskietce.
Użycie parametru /b powoduje stworzenie na dyskietce miejsca na zbiory systemowe dowolnej wersji DOSu.
Parametru /b nie można używać razem z parametrami /s i /v.
Użycie parametru /4 umożliwia formatowanie dyskietek na 360 KB w napędach 1.2 MB. Należy liczyć się z tym, że ze względów technicznych nogą wystąpić kłopoty z odczytaniem tak sformatowanych dyskietek w napędach 360 KB.

HDD:
W przypadku użycia polecenia **FORMAT** do formatowania twardego dysku (np. C:) zostanie wyprowadzony następujący komunikat:

```
Warning, All Data On Non-Removable
Disk Drive c: Will Be Lost!
Proceed with Format (Y/N)?
```

czyli

```
Ostrzeżenie, wszystkie dane
z twardego dysku c: zostaną stracone!
Kontynuować wykonanie Format (T/N)?
```

Jeżeli rozpoczęliśmy formatowanie twardego dysku przez pomyłkę, należy wcisnąć n i ENTER. Wciśnięcie y i ENTER spowoduje sformatowanie twardego dysku (taka operacja może trwać nawet kilka minut).
□

Polecenie **FORMAT** ignoruje działanie polecenia **ASSIGN**, lecz nie można stosować **FORMAT** do napędów objętych działaniem **JOIN** i **SUBST**.
Działanie polecenia **FORMAT** zostało omówione szczegółowo w części "Kopiowanie dyskietek i zbiorów" niniejszego rozdziału.

GRAFTABL

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**GRAFTABL**

Funkcja: załadowanie dodatkowych znaków graficznych do pamięci komputera. Polecenie obsługuje sprzęt tylko z kartami CGA.

GRAPHICS

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**GRAPHICS** [*typ drukarki*][/r][/b]

Funkcja: umożliwia przeniesienie obrazu z ekranu podczas pracy w trybie graficznym na drukarkę przez jednoczesne wciśnięcie klawiszy Shift i PrtSc (ang. print screen, czyli drukuj ekran). Polecenie współpracuje tylko z kartą CGA. Obsługiwane są następujące typy drukarek:

COLOR1 - drukarki IBM PCCP (Personal Computer Color Printer) z czarną taśmą i zgodne.
COLOR4 - drukarki IBM PCCP z taśmą RGB (red-green-blue, czyli czerwono-zielono-niebieską) i zgodne.
COLOR8 - drukarki IBM PCCP z taśmą CMY (cyan-magenta-yellow, czyli cyjanowo-magentowo-żółtą) i zgodne.
COMPACT - drukarki IBM Personal Computer Compact Printer i zgodne.
GRAPHICS - drukarki IBM Personal Graphics Printer i zgodne.

Standardowo typ drukarki jest ustawiany na GRAPHICS i to w zupełności wystarcza, ponieważ większość drukarek jest zgodna z tym właśnie typem drukarek IBM.
Użycie parametru /r powoduje drukowanie w pełni zgodne z obrazem na ekranie monitora. Jeżeli na ekranie mamy białe litery na czarnym tle, to na wydruku również otrzymamy białe litery na czarnym tle. Standardowo są drukowane czarne litery na białym tle, a więc odwrotnie, niż na ekranie.
Użycie parametru /b (tylko z COLOR4 i COLOR8) powoduje wydruk tła.
Należy pamiętać o tym, że polecenie **GRAPHICS** jest rezydentne i zwiększa obszar pamięci zajmowany na stałe przez DOS - na to powinni zwrócić uwagę użytkownicy systemów z niewielką pamięcią operacyjną i programów wymagających dużej pamięci operacyjnej.

JOIN

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**JOIN**
lub
[*d:*][*path*]**JOIN** *d:* *d:\katalog*
lub
[*d:*][*path*]**JOIN** *d:* /d

Funkcja: połączenie napędu dysków z katalogiem w innym napędzie dysków w celu utworzenia pojedynczej struktury katalogu. Jeżeli chcemy np. połączyć napęd A: z katalogiem \KATB na dyskietce w napędzie B:, to należy podać polecenie

A:\=>join a: b:\katb <ENTER>

Katalog \KATB musi być pusty, aby polecenie **JOIN** zadziałało. Jeżeli w napędzie B: nie ma katalogu \KATB, to zostanie on stworzony przez **JOIN**. Nie można używać **JOIN** z ani z katalogiem głównym, ani z katalogiem aktualnym jako parametrem. Podanie polecenia **JOIN** bez parametrów:

A:\=>join <ENTER>

spowoduje wyprowadzenie informacji o napędach i katalogach aktualnie z nimi połączonych. W naszym przypadku będzie to

A: => B:\KATB

Ponieważ napęd A: jest połączony z katalogiem \KATB w napędzie B:, nie jest możliwe wydawanie jakichkolwiek poleceń odnoszących się do napędu A:. W celu odłączenia katalogu głównego w napędzie A: od katalogu \KATB w B: należy podać polecenie

A:\=>join a:/d <ENTER>

Użycie parametru /d (ang. disconnect, czyli odłącz) powoduje usunięcie połączenia katalogu głównego wyspecyfikowanego napędu od katalogu, z którym połączenie było aktywne.
Przy aktywnym poleceniu **JOIN** nie można używać poleceń **BACKUP, DISKCOPY, DISKCOMP, FORMAT** i **RESTORE.**

KEYBxx

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]KEYBUK
lub
[*d:*][*path*]KEYBGR
lub
[*d:*][*path*]KEYBFR
lub
[*d:*][*path*]KEYBIT
lub
[*d:*][*path*]KEYBSP

Funkcja: wczytanie rezydentnego programu redefiniującego klawiaturę zgodnie z normami obowiązującymi w:

- Wielkiej Brytanii,
- R.F.N.,
- Francji,
- Włoszech,
- Hiszpanii.

LABEL

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]LABEL [*d:*][*nazwa dysku*]

Funkcja: nadanie, usunięcie lub zmiana nazwy dysku (dyskietki). Polecenie to nie może być używane, jeżeli są aktywne **ASSIGN** lub **SUBST**.

LINK

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**LINK**

Funkcja: tworzenie ze zbiorów wynikowych i bibliotecznych zbiorów zawierających programy wykonywalne. Dodatkowo są tworzone zbiory zawierające wykaz nazw, adresów i długości wszystkich segmentów utworzonego programu wykonywalnego (tzw. zbiory mapowe).
Z konsolidatorem (ang. linker) dostarczanym z PC DOS można pracować w trybie konwersacyjnym i dwóch rodzajach trybu wsadowego. Poniżej zostanie omówiony, jako najprostszy, tryb pracy konwersacyjnej. Podanie polecenia

A:\=>link <ENTER>

powoduje rozpoczęcie pracy konsolidatora w trybie konwersacyjnym. Zostaje wyświetlony komunikat

Object Modules [.OBJ]:

czyli

Moduły wynikowe [.OBJ]:

będący zapytaniem o moduły wynikowe. Domniemane rozszerzenie nazw zbiorów zawierających moduły wynikowe to pokazane w nawiasach kwadratowych OBJ. Należy podać nazwy zbiorów zawierających moduły wynikowe (utworzone wcześniej za pomocą jakiegoś kompilatora) i wcisnąć klawisz ENTER. Odpowiedzią będzie komunikat

Run File [*nzwzbru*.EXE]:

czyli

Zbiór wykonywalny [*nzwzbru*.EXE]:

Nzwzbru to pierwsza nazwa zbioru zawierającego moduły wynikowe. Zatem *nzwzbru*.EXE to proponowana nazwa zbioru zawierającego program wykonywalny. Jeżeli nazwę tę akceptujemy, należy wcisnąć klawisz ENTER. Jeżeli nie, należy wpisać inną nazwę i zakończyć jej wprowadzanie również wciśnięciem ENTER. Na ekranie pojawi się kolejny komunikat:

List File [NUL.MAP]:

czyli

Zbiór mapowy [NUL.MAP]:

Akceptujemy proponowaną nazwę zbioru, który będzie zawierał mapę zbioru (nazwy, adresy, długości segmentów) z programem wykonywalnym, lub podajemy swoją nazwę. Następnie zostaje wyświetlony komunikat

Libraries [.LIB]:

czyli

Zbiory biblioteczne [.LIB]:

Podajemy (jeżeli jest to potrzebne) nazwy zbiorów bibliotecznych i wciskamy ENTER. W ten sposób rozpocznie się proces tworzenia zbioru zawierającego program wykonywalny bezpośrednio pod nadzorem systemy operacyjnego.
Podczas pracy z konsolidatorem należy pamiętać, że wszystkie podawane nazwy zbiorów muszą zostać podane wraz ze ścieżkami dostępu, jeżeli zbiory te nie znajdują się w aktualnym katalogu. W przypadku podawania nazw więcej, niż jednego zbioru określonego typu (np. kilka zbiorów bibliotecznych), nazwy tych zbiorów powinny być połączone znakiem plus (+). Jeżeli wszystkie nazwy zbiorów określonego typu nie mieszczą się w jednym wierszu, jako ostatni znak w danym wierszu należy umieścić plus (+). Po wciśnięciu klawisza ENTER będzie można dopisać brakujące nazwy zbiorów.
Podczas pracy z konsolidatorem w trybie konwersacyjnym, na

dowolnym etapie tej pracy można podać parametry sterujące przebiegiem tworzenia programu wykonywalnego. Oto niektóre z tych parametrów:

/he - powoduje wyprowadzenie na ekran listy dozwolonych parametrów. Ten parametr nie może zostać podany za wprowadzaną nazwą zbioru.

/p - przed zapisaniem zbioru z programem wykonywalnym na dysk użytkownik zostanie poproszony o zmianę dyskietki.

/e - powoduje optymizację tworzonego kodu wykonywalnego (kod ten nie będzie akceptowany przez debuggery symboliczne, np. Microsoft Symbolic Debug Utility, tzw. SYMBDEB).

/m - powoduje utworzenie rozszerzonego zbioru typu mapowego (o nazwie zgodnej z nazwą pierwszego zbioru wynikowego i rozszerzeniu MAP), pomocnego w przypadku późniejszego korzystania z debuggerów symbolicznych.

/noi - powoduje rozróżnienie dużych i małych liter w kodzie wynikowym.

/nod - powoduje zignorowanie wszelkich informacji o zbiorach bibliotecznych, które należy dołączyć, zawartych w kodzie wynikowym, a generowanych przez niektóre kompilatory. Użytkownik zyskuje wtedy pewność, że zostaną dołączone tylko te zbiory biblioteczne, których nazwy podano jawnie podczas pracy z konsolidatorem.

/h - w utworzonym programie wykonywalnym adres startowy zostanie ustawiony na możliwie najwyższy dostępny w pamięci (bez parametru /h adres ten jest ustawiany na możliwie najniższą wartość).

Przykład użycia konsolidatora:

```
A:\=>link <ENTER>
Object Modules [.OBJ]: baza1+baza2+baza3+ <ENTER>
Object Modules [.OBJ]: baza4/p <ENTER>
Run File [baza1.EXE]: <ENTER>
List File [NUL.MAP]: mapa <ENTER>
Libraries [.LIB]: c:\nantuc\clip <ENTER>
```

Wynikiem tak przeprowadzonej sesji z konsolidatorem systemowym będzie konsolidacja zbiorów BAZA1.OBJ, BAZA2.OBJ, BAZA3.OBJ i BAZA4.OBJ, zawierających moduły wynikowe; wyszukanie zbioru bibliotecznego CLIP.LIB na dysku C: w katalogu \NANTUC; utworzenie zbioru BAZA1.EXE, zawierającego program wykonywalny. Przed zapisaniem tego zbioru na dysk zostanie wyprowadzony komunikat z prośbą o zmianę dyskietki w napędzie A: (jest to spowodowane użyciem parametru /p).
Ostatnio coraz większą popularność (głównie dzięki nieocenionej firmie Borland) zdobywają kompilatory i konsolidatory pracujące w tzw. środowisku zintegrowanym. Dla użytkownika proces przekształcenia kodu źródłowego w kod wykonywalny jest procesem spójnym (uruchamianym przez jedno wciśnięcie klawisza i dalej przebiegającym automatycznie), niemożliwe jest odróżnienie np. etapu kompilacji od etapu konsolidacji. Dlatego takie produkty, jak powyżej omówiony konsolidator, są używane coraz niechętniej i powoli odchodzą w zapomnienie.

MKDIR lub MD

Polecenie wewnętrzne.

Składnia: **MD** [*d:*]*path*
lub
MKDIR [*d:*]*path*

Funkcja: tworzenie katalogów. Należy pamiętać o tym, że całkowita długość ścieżki dostępu do najniżej położonego katalogu nie może przekraczać 63 znaków. Podanie polecenia

```
A:\POZIOM1\POZIOM2=>md poziom3 <ENTER>
```

spowoduje utworzenie katalogu POZIOM3 o ścieżce dostępu spod katalogu głównego \POZIOM1\POZIOM2\POZIOM3. Natomiast polecenie

```
A:\POZIOM1\POZIOM2=>md\poziom3 <ENTER>
```

spowoduje utworzenie katalogu będącego bezpośrednio podkatalogiem katalogu głównego. Przyczyną takiego zadziałania polecenia **MD** jest znak \ (backslash). Wskazuje on na konieczność

rozpoczęcia tworzenia nowego podkatalogu z poziomu katalogu głównego. Spacja użyta w poprzednim poleceniu powoduje utworzenie podkatalogu z poziomu katalogu aktualnego.
Struktura, przeznaczenie i właściwości katalogów zostały omówione w części "Praca z katalogami".
Nie należy używać **MD** z aktywnymi poleceniami **ASSIGN**, **JOIN** i **SUBST**.

MODE

Polecenie zewnętrzne.

Składnia: 1) [*d:*][*path*]MODE LPT#[:][*n*][,[*m*][,p]]

lub

2) [*d:*][*path*]MODE *n*

lub

[*d:*][*path*]MODE [*n*],*m*[,t]

lub

3) [*d:*][*path*]MODE COMn[:]*liczbabodów*
[,[*parzystość*][,[*bitydanych*][,[*bitystopu*]
[,p]]]]

lub

4) [*d:*][*path*]MODE LPT#[:]=COM*n*

Funkcja: ustawienie parametrów działania drukarki, monitora współpracującego z kartą CGA lub łącza komunikacji szeregowej. Polecenie ma cztery tryby pracy:

1) Ustawianie trybu pracy drukarki.

\# - oznacza numer drukarki, od 1 do 3,
n - jest ilością znaków w wierszu, 80 lub 132 (standardowo 80),
m - określa odstęp między wierszami (w wierszach na cal), 6 lub 8 (standardowo 6),
p - powoduje ponawianie prób wysyłania danych na drukarkę w przypadku pojawienia się błędu.

2) Ustawianie trybu pracy karty CGA i monitora.

W miejsce n możemy wstawić następujące wartości:
40 - ustawienie na 40 znaków w wierszu,
80 - ustawienie na 80 znaków w wierszu,

BW40 - ustawienie na 40 znaków w wierszu w trybie czarno-białym,

BW80 - ustawienie na 80 znaków w wierszu w trybie czarno-białym,

CO40 - ustawienie na 40 znaków w wierszu w trybie kolorowym,

CO80 - ustawienie na 80 znaków w wierszu w trybie kolorowym,

MONO - przełączenie w tryb MDA (monochromatyczny tryb znakowy).

Parametr *m* może przybrać wartość r lub l. Wstawienie r powoduje przesunięcie obrazu o jeden (przy 40 znakach we wierszu) lub dwa (przy 80 znakach we wierszu) znaki w prawo, wstawienie l powoduje analogiczne przesunięcie obrazu w lewo.
Użycie parametru t powoduje przeprowadzenie testu pozwalającego na właściwe ustawienie obrazu na monitorze.

3) Ustawianie sposobu pracy łącza komunikacji szeregowej.

Można określić następujące parametry protokołu komunikacji:

n - numer portu, 1 lub 2.

liczbabodów - określa szybkość transmisji. Dozwolone wartości: 110, 150, 300, 600, 1200, 2400, 4800 i 9600. Z liczby określającej szybkość transmisji są pobierane tylko dwie pierwsze cyfry i one wystarczają do zdeterminowania wartości całego parametru.

parzystość - możemy ją ustawić na N (ang. none, transmisja danych nie jest weryfikowana), O (ang. odd, uzupełnienie do nieparzystości) i E (ang. even, uzupełnienie do parzystości, wartość ustawiana standardowo).

bitydanych - transmisja 7- lub 8-bitowa (podajemy 7 lub 8, standardowo ustawione 7).

bitystopu - 2 (ustawiane standardowo dla 110 bodów) lub 1 (ustawiane standardowo dla innych szybkości transmisji).

p - ponowna próba transmisji po błędzie.

4) Skierowanie przesyłu danych do łącza komunikacji szeregowej zamiast do równoległego portu drukarki.

- określa numer portu drukarki (1, 2 lub 3),

n - określa numer łącza komunikacji szeregowej (1 lub 2).

MORE

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]MORE

Funkcja: pobranie danych z urządzenia wejścia i wyprowadzenie ich na ekran. Po wypełnieniu całego ekranu jest wyświetlany komunikat --More--, a wyprowadzanie danych na ekran jest wstrzymane do momentu naciśnięcia jakiegokolwiek klawisza przez użytkownika. Przykłady zastosowań:

```
A:\=>more < spis.srt <ENTER>
```

```
A:\=>tree | more <ENTER>
```

PATH

Polecenie wewnętrzne.

Składnia: **PATH** [[*d:*]*path*[[;[*d:*]*path*]...]]

lub

PATH;

Funkcja: podanie ścieżki (ścieżek) dostępu do katalogów, w których powinny być szukane zbiory programowe nie znalezione w katalogu aktualnym. Szukane są tylko programy o rozszerzeniach nazwy COM, EXE i BAT. Załóżmy, że program DT znajduje się w katalogu \SYSUT\NORTONY na dysku C:, a program PCT.EXE w katalogu \SYSUT. Katalogiem aktualnym jest katalog główny na C:. Podanie ścieżki

```
C:\=>path c:\sys;c:\sysut\nortony;a:. <ENTER>
```

spowoduje, że w przypadku podania polecenia

```
C:\=>dt <ENTER>
```

program DT.COM zostanie znaleziony i wykonany. Natomiast polecenie

```
C:\=>pctools <ENTER>
```

zostanie skwitowane komunikatem

Bad command or filename

ponieważ ani w katalogu aktualnym, ani w żadnym z katalogów wymienionych w poleceniu **PATH** nie ma wykonywalnego programu o nazwie PCTOOLS. Polecenie

A:\=>path <ENTER>

wyświetla aktualnie podane ścieżki dostępu, a

A:\=>path; <ENTER>

sprawia, że DOS "zapomni" o ścieżkach podanych do tej pory.

PRINT

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**PRINT** [/d:*n1*][/b:*n2*][/u:*n3*] [/m:*n4*][/s:*n5*][/q:*n6*][/c][/t][/p][*d:*] [*path*][*nzwzbru*][.*roz*]...]

Funkcja: wysyła na drukarkę wyspecyfikowane zbiory. Transmisja danych odbywa się niezauważalnie dla użytkownika - oznacza to, że po rozpoczęciu i w trakcie drukowania system jest gotowy do dalszej pracy. Nie jest wtedy jednak możliwe uzyskanie dostępu do drukarki z innych programów czy za pomocą innych poleceń. Parametry polecenia **PRINT**:

/d:*n1* - określa urządzenie, na które ma odbyć się transmisja (standardowo PRN). Parametr ten musi być podany jako pierwszy. Jako *n1* można podać LPT1, LPT2, LPT3, COM1 COM2 i AUX,

/b:*n2* - określa w bajtach rozmiar wewnętrznego bufora (standardowo 512),

/q:*n3* - określa maksymalną ilość zbiorów w kolejce do drukowania (od 1 do 32, standardowo 10),

/s:*n4* - określa czas, przez jaki są pobierane porcje danych do drukowania (od 1 do 255, standardowo 8),

/u:*n5* - określa liczbę taktów zegara, przez którą polecenie **PRINT** czeka na dostęp do drukarki (standardowo 1). Po upływie *n5* taktów zegara **PRINT** rezygnuje z pobrania danych do drukowania,

/m:*n6* - określa liczbę taktów zegara, w czasie której **PRINT** wysyła dane na drukarkę (od 1 do 255, standardowo 2)

/t - przerywa działanie **PRINT** (wstrzymanie drukowania i usunięcie wszystkich zbiorów z kolejki),

/c - usuwa z kolejki zbiór poprzedzający parametr /c i wszystkie następne zbiory aż do napotkania parametru /p lub znaku końca linii (<ENTER>),

/p - dołącza do kolejki zbiór poprzedzający parametr /p i wszystkie następne zbiory aż do napotkania parametru /c lub znaku końca linii (<ENTER>).

Parametry /b, /d, /m, /q i /s mogą zostać podane tylko podczas pierwszego użycia polecenia **PRINT**. Przy parametrze /d nie można podać nazwy urządzenia nie podłączonego do komputera (może to wywołać nieoczekiwane zachowanie się systemu).
Przykład zastosowania polecenia **PRINT**:

```
A:\=>print /d:lpt2 /b:2048 /q:15 zbrnr*.txt <ENTER>
```

Taka forma polecenia **PRINT** powoduje wydrukowanie kolejno zbiorów z grupy ZBRNR*.TXT na drukarce LPT2; wielkość bufora: 2 KB; maksymalna ilość zbiorów w kolejce: 15.

PROMPT

Polecenie wewnętrzne.

Składnia: **PROMPT** [*tekst-zgłoszenia*]

Funkcja: zmiana znaku zgłoszenia się systemu. Zamiast standardowego znaku gotowości pojawia się tekst wprowadzony jako parametr polecenia **PROMPT**.
Znak **$** i umieszczony za nim znak dodatkowy pozwalają na umieszczenie w znaku gotowości systemu następujących informacji:

- daty ($d),
- czasu ($t),
- ścieżki dostępu do aktualnego katalogu ($p),
- numeru wersji DOSu ($v),
- oznaczenia aktualnego napędu ($n),
- znaku > ($g),
- znaku < ($l),
- znaku | ($b),

- znaku = ($q),
- znaku backspace ($h),
- znaku dolara ($$),
- znaku ESCape ($e),
- znaków CR i LF ($_) czyli zmiany wiersza,

Polecenie

A> prompt pq$g <ENTER>

daje znak zgłoszenia zawierający informację o napędzie i katalogu aktualnym, znak równości i znak większości. Jeżeli katalogiem aktualnym jest \SYS na napędzie C:, to znak zgłoszenia się systemu będzie wyglądał następująco:

C:\SYS=>

RECOVER

Polecenie zewnętrzne.

Składnia: 1) [*d:*][*path*]**RECOVER** [*d:*][*path*]*nzwzbru*[.*roz*]
lub
2) [*d:*][*path*]**RECOVER** *d:*

Funkcja: odczytywanie zbioru (zbiorów) z uszkodzonego dysku (dyskietki). Opcja 1 pozwala na odczytanie wskazanego zbioru, opcja 2 - na odczytanie wszystkich zbiorów ze wskazanego dysku. Dane z uszkodzonych sektorów nie są odczytywane.

RENAME lub **REN**

Polecenie wewnętrzne.

Składnia: **REN** [*d:*][*path*]*nzwzbru*[.*roz*] *nzwzbru*[.*roz*]

Funkcja: zmiana nazwy zbioru (nazw gupy zbiorów). Nazwa wymieniona jako pierwsza jest zastępowana przez drugą nazwę. W obydwu nazwach można stosować znaki blankietowe.

RESTORE

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]**RESTORE** *d:* [*d:*][*path*] *nzwzbru*[.*roz*]
[/s][/p]

Funkcja: polecenie służy do przywrócenia kopii rezerwowych zapisanych na dysku poleceniem **BACKUP** do stanu wyjściowego. Pierwsze oznaczenie stacji dysków (*d:*) wskazuje na stację z dyskietką (dyskiem) zawierającą kopie rezerwowe, natomiast [*d:*][*path*]*nazwabru*[.*roz*]wskazuje na to, dokąd i jakie zbiory chcemy reaktywować.

Użycie parametru /s oznacza, że reaktywowanie kopii rezerwowych ma dotyczyć również kopii wykonanych ze zbiorów w podkatalogach.

Użycie parametru /p powoduje, że przed reaktywowaniem zbioru oznaczonego jako "read-only" lub zmienianego po ostatnim kopiowaniu go poleceniem **BACKUP** jest wyprowadzany komunikat o zaistniałej sytuacji i użytkownik zostaje zapytany, czy dalej życzy sobie reaktywowania zbioru. Parametr ten winien być użyty, jeżeli kopie rezerwowe zostały utworzone za pomocą wcześniejszej wersji DOSu, niż aktualnie używana. Przypadkowe reaktywowanie zbiorów IBMBIO.COM i IBMDOS.COM mogłoby spowodować zniszczenie zbiorów systemowych wersji późniejszej (stare zbiory zostałyby zapisane na nowych), a parametr /p daje nam możliwość usunięcia niepożądanych kopii rezerwowych z listy do reaktywowania.

Po zakończeniu działania polecenie **RESTORE** ustawia parametr systemowy ERRORLEVEL na

0 - jeżeli reaktywowanie przebiegło poprawnie,
1 - jeżeli nie zostały znalezione żadne zbiory, które można by było reaktywować,
2 - jeżeli niektóre zbiory nie zostały reaktywowane (np. ze względu na konflikt podziału podczas pracy komputera w sieci),
3 - jeżeli wykonywanie polecenia zostało przerwane przez użytkownika (<Ctrl>-<C>),
4 - jeżeli wykonanie polecenia nie było możliwe.

Nie należy stosować polecenia **RESTORE**, jeżeli podczas działania **BACKUP** było aktywne jedno z poleceń **ASSIGN**, **JOIN** lub **SUBST**. Podanie polecenia

```
A:\=>restore b: a:\*.txt /s <ENTER>
```

powoduje odtworzenie z dysku B: wszystkich zbiorów o rozszerzeniu nazwy TXT na dysk A:. Odtwarzane są także te kopie,

które były wykonane ze zbiorów znajdujących się w katalogach poziomu niższego od katalogu głównego.

RMDIR lub RD

Polecenie wewnętrzne.

Składnia: **RMDIR** [*d:*]*path*

lub

RD [*d:*][*path*]

Funkcja: usunięcie katalogu z dysku. Nie jest możliwe usunięcie katalogu głównego ani aktualnego połączonego z innym katalogiem poleceniem **SUBST**. Katalog, który ma zostać usunięty, nie może zawierać w sobie żadnych zbiorów (poza . i ..).

SELECT

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]SELECT xxx yy

Funkcja: utworzenie dyskietki systemowej za pomocą **DISKCOPY** oraz generacja zbiorów AUTOEXEC.BAT i CONFIG.SYS zawierających informacje o dopasowaniu układu klawiatury do standardu obowiązującego w jednym z następujących krajów:

Kraj	xxx	yy
USA	001	US
Francja	033	FR
Hiszpania	034	SP
Włochy	039	IT
Wielka Brytania	044	UK
RFN	049	GR

Parametr xxx określa format daty, oznaczenie waluty i separator dziesiętny.
Parametr yy określa układ klawiatury.

SET

Polecenie wewnętrzne.

Składnia: SET [*parametr*=[*łańcuch_znaków*]]

Funkcja: polecenie jest używane do definiowania parametrów środowiskowych systemu operacyjnego.

Jeżeli podamy polecenie

```
A:\=>set operator=ziuta <ENTER>
```

to przypisze ona nazwie zmiennej %OPERATOR% łańcuch znaków "ZIUTA". Zmienna ta może być wykorzystana np. w zbioru MOJKATAL.BAT o zawartości

```
cd\%operator%
dir/p
```

Podanie polecenia

```
A:\=>mojkatal <ENTER>
```

spowoduje zmianę katalogu aktualnego na \ZIUTA (zmienna %OPERATOR% została zastąpiona przez łańcuch znaków ZIUTA) i wyświetlenie zawartości katalogu. Jeżeli podana zostanie polecenie

```
A:\=>set operator=ciapulek <ENTER>
```

to wynikiem zadziałania polecenia

```
A:\=>mojkatal <ENTER>
```

będzie zmiana katalogu aktualnego na \CIAPULEK i wyświetlenie jego zawartości.
W poleceniu SET może być użyty dowolny łańcuch znaków nie zawierający cyfr.
Podanie polecenia SET bez parametrów powoduje wyprowadzenie informacji o aktualnej wartości trzech zmiennych systemowych: COMSPEC, PATH i PROMPT. Użycie SET do zmiany wartości zmiennej COMSPEC zostało opisane w następnej części niniejszego rozdziału.
Jeżeli polecenie SET zostanie podane w postaci SET *nazwa*, to spowoduje to anulowanie wartości uprzednio przypisanej do zmiennej *nazwa*.

SORT

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]SORT [/r][/+*n*]

Funkcja: wczytanie danych z urządzenia wejścia, posortowanie ich i wysłanie na standardowe urządzenie wyjścia. Sortowanie odbywa się według wartości znaków w kodzie ASCII z jednym wyjątkiem - duże i małe litery nie są rozróżniane przez SORT. Maksymalna wielkość sortowanego zbioru: 63 KB. Jeżeli wyniki sortowania mają zostać wpisane do zbioru, to nazwa zbioru wyjściowego musi być inna niż zbioru wejściowego.
Użycie parametru /r sprawia, że sortowanie odbywa się zstępująco (np. z jest przed a).
Parametr /+*n* określa numer kolumny zbioru, od której ma zacząć się sortowanie (wartością standardową jest kolumna pierwsza). Linie zbioru po posortowaniu będą ustawione zgodnie z kolejnością znaków w *n*-tej kolumnie.

SUBST

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]SUBST *d:* *d:path*
lub
[*d:*][*path*]SUBST *d:* /d
lub
[*d:*][*path*]SUBST

Funkcja: polecenie to przyporządkowuje nazwę *d:* do istniejącego katalogu *d:path*. Takie przyporządkowanie umożliwia pracę z aplikacjami nie dostosowanymi do obsługi struktur katalogowych.
Użycie parametru /d powoduje przywrócenie stanu sprzed podania polecenia SUBST.
Podanie samego polecenia [*d:*][*path*]SUBST powoduje wyświetlenie aktualnie aktywnego przyporządkowania.

SYS

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]SYS *d:*

Funkcja: przesłanie zbiorów systemowych IBMBIO.COM i IBMDOS.COM z dyskietki w napędzie aktualnym na dyskietkę w na-

pędzie wyspecyfikowanym przez *d:*. Katalog dysku docelowego musi być pusty, aby przesłanie było możliwe.

TIME

Polecenie wewnętrzne.

Składnia: **TIME** [*gg*:*mm*:[:*ss*[.*xx*]]]

Funkcja: odczyt i ustawienie czasu. Polecenie **TIME** podane bez parametrów powoduje wyświetlenie komunikatu

Current time is *gg*:*mm*:*ss*.*xx*
Enter new time:

czyli

Aktualny czas *gg*:*mm*:*ss*.*xx*
Podaj nowy czas:

Oznaczenie *gg* odnosi się do godzin, *mm* do minut, *ss* do sekund, *xx* do setnych sekundy. Nowy czas podajemy, wpisując godziny, minuty i sekundy rozdzielone dwukropkami, na przykład 12:34:30. Minut i sekund nie trzeba koniecznie podawać - jeżeli nie napiszemy nic w miejscu przewidzianym na podanie minut i sekund, to system sam ustawi je na zero. Czas przez nas podany zacznie być odmierzany dopiero po wciśnięciu klawisza ENTER. Polecenie **TIME** można też podać bezpośrednio w postaci

A:\=>time *gg*:*mm*:*ss*.*xx* <ENTER>

Format podawania czasu może być zmieniony przez zastosowanie poleceń **COUNTRY** lub **SELECT**.

TREE

Polecenie zewnętrzne.

Składnia: [*d:*][*path*]TREE [*d:*][/f]

Funkcja: podanie wszystkich ścieżek dostępu istniejących na dyskietce (dysku) w wyspecyfikowanym napędzie.
Użycie prametru /f powoduje wyświetlenie nazw zbiorów zawartych w każdym z katalogów.

Zaleca się używanie polecenia **TREE** z filtrem **MORE**, np.

```
A:\=>tree c:/f | more <ENTER>
```

TYPE

Polecenie wewnętrzne.

Składnia: **TYPE** [*d:*][*path*]*nzwzbru*[*.roz*]

Funkcja: wyprowadzenie wyspecyfikowanego zbioru na standardowe urządzenie wyjścia. Np. polecenie

```
A:\=>type mojzbr.txt > prn <ENTER>
```

powoduje wyprowadzenie zbioru MOJZBR.TXT na drukarkę.

VER

Polecenie wewnętrzne.

Składnia: **VER**

Funkcja: wyświetla na standardowym urządzeniu wyjścia komunikat o stosowanej wersji DOSu, np.

```
A:\=>ver <ENTER>

IBM Personal Computer DOS Version 3.10
```

VERIFY

Polecenie wewnętrzne.

Składnia: **VERIFY** [**ON** | **OFF**]

Funkcja: weryfikuje prawidłowość zapisu danych na dysk. Ustawienie **VERIFY** na **ON** powoduje spowolnienie pracy systemu podczas wszelkich operacji zapisywania danych.
Podanie polecenia **VERIFY** bez parametrów wyświetla informację o aktualnym stanie weryfikacji (włączona / wyłączona).

VOL

Polecenie wewnętrzne.

Składnia: **VOL** [*d:*]

Funkcja: wyprowadzenie informacji o nazwie dyskietki (dysku) w wyspecyfikowanym napędzie.

3.9 Konfiguracja systemu

DOS pozwala na ustawienie wielu parametrów pracy i konfiguracji systemu zgodnie z potrzebami użytkownika. Informacje o tych parametrach są podawane w zbiorze CONFIG.SYS. Każde polecenie stanowi osobny wiersz tego zbioru. Zawartość CONFIG .SYS jest czytana przez system po każdym zimnym lub gorącym starcie i tylko wtedy informacje zawarte w tym zbiorze są uwzględniane przy ustalaniu ostatecznego kształtu środowiska systemowego. Nie jest możliwe zatem wykonanie instrukcji zawartych w CONFIG.SYS podczas pracy systemu - CONFIG.SYS najpierw trzeba zapisać w katalogu głównym dysku, z którego jest wczytywany system, a potem ponownie uruchomić sprzęt, np. za pomocą "gorącego startu" (jednoczesne wciśnięcie klawiszy Ctrl, Alt i Del), co spowoduje uwzględnienie parametrów pracy systemu podanych w CONFIG.SYS.

Poniżej zostały opisane polecenia, które można zawrzeć w CONFIG.SYS.

BREAK

To polecenie zostało omówione w poprzedniej części rozdziału 3 ("Polecenia DOSu"). W zbiorze CONFIG.SYS polecenie to ma następującą postać:

BREAK ON|OFF

BUFFERS

Polecenie **BUFFERS** określa liczbę buforów, które będą używane przez system i jest podawane w postaci

BUFFERS = *n*

gdzie n jest liczbą buforów. Bufor oznacza pewien obszar (blok) pamięci RAM, który przez system jest wykorzystywany podczas pracy ze zbiorami czytanymi z dysku. Jeżeli jakiś program aplikacyjny żąda od DOSu przeczytania zbioru z dysku, to DOS po przeczytaniu tego zbioru umieszcza jego fragment

(przeważnie jest to 128-bajtowy rekord) w jednym z buforów i oznacza bufor jako wykorzystywany. Podczas następnego żądania dostępu do zbioru są sprawdzane najpierw wykorzystywane bufory, a dopiero potem dysk. Pobranie fragmentu zbioru i przekazanie go z bufora programowi aplikacyjnemu pochłania znacznie mniej czasu, niż przeprowadzenie takiej operacji z dyskiem. Tak więc kreowanie większej liczby buforów może przyspieszyć pracę z aplikacjami często odwołującymi się do wielu różnych zbiorów (są to przede wszystkim bazy danych i tzw. pakiety zintegrowane) - zwiększa się wtedy prawdopodobieństwo znalezienia potrzebnego fragmentu zbioru w którymś z buforów. Natomiast stosowanie wielu buforów w przypadku korzystania z programów współpracujących z dyskiem niezbyt intensywnie nie ma większego sensu, bowiem przeszukanie dużej liczby buforów zajmie więcej czasu, niż przeczytanie zbioru od razu z dysku. Należy też pamiętać o tym, że sprzęt z niewielką pamięcią operacyjną (np.128 KB lub 256 KB) źle znosi otwieranie wielu buforów - jest to związane z wielkością pojedynczego bufora (512 bajtów) i zwiększaniem przez każdy otwarty bufor rezydentnej części DOSu o 528 bajtów. Wiele otwartych buforów oznacza zmniejszenie obszaru RAMu dostępnego dla oprogramowania.

Wartości *n* mieszczą się w przedziale od 1 do 99, standardowo (jeżeli nie jest to zmienione poleceniem **BUFFERS**) są kreowane dwa bufory (n=2). Dla baz danych pracujących z podkatalogami zaleca się utworzenie od 10 do 25 buforów.

HDD:
Posiadacze twardych dysków powinni używać co najmniej 3 buforów, nawet jeżeli wykorzystywane oprogramowanie nie współpracuje intensywnie z dyskiem.
□

COUNTRY

Polecenie służy do określenia symbolu waluty, separatora dziesiętnego oraz formatu czasu i daty, jaki chcemy stosować. Ma ono postać

COUNTRY *xxx*

gdzie xxx jest kodem państwa. Kody państw zostały podane w poprzedniej części rozdziału 3 ("PoleceniaDOSu") przy opisie polecenia SELECT. Należy zauważyć, że SELECT dodaje do zbioru CONFIG.SYS na tworzonym dysku systemowym polecenie **COUNTRY** wraz z odpowiednim parametrem.

DEVICE

Polecenie to powoduje wczytanie przez DOS sterownika programowego (ang. installable device driver) określonego urządzenia. Sterownik programowy jest krótkim programem (napisanym przeważnie w asemblerze, czyli języku symbolicznym), którego zadaniem jest pośredniczenie między DOSem a urządzeniem zewnętrznym. Konstruktorom trudno jest budować takie urządzenia, które swoje wyjście i wejście miałyby dostosowane tylko do wymagań DOSu - dlatego drajwery (takie określenie sterowników programowych też jest stosowane) dokonują czegoś w rodzaju tłumaczenia z języka DOSu na język urządzenia i odwrotnie.
Polecenie **DEVICE** jest podawane w postaci

DEVICE [*d:*][*path*]*nzwzbru*[*.roz*]

gdzie *nzwzbru*[*.roz*] jest nazwą zbioru będącego drajwerem.

W czasie uruchamiania DOS sam wczytuje swoje standardowe sterowniki (do obsługi klawiatury, monitora, stacji dysków, drukarki itd.) i nie musimy się nimi zajmować. Mamy jednak możliwość wykorzystania dodatkowych sterowników dostarczanych razem z systemem operacyjnym. Są to ANSI.SYS i VDISK.SYS.

ANSI.SYS jest rozszerzoną wersją standardowego sterownika klawiatury i ekranu. Pozwala na przypisywanie łańcuchów znaków poszczególnym klawiszom (czyli definiowanie znaczenia każdego z nich) i umożliwia pełne kontrolowanie atrybutów (parametrów) pracy ekranu. W następnym rozdziale zastosowanie ANSI.SYS zostanie omówione bardziej szczegółowo - teraz podamy jedynie kody przyporządkowane klawiszom przez ANSI.SYS oraz ciągi znaków sterujących, używanych przy wykorzystaniu ANSI.SYS. Każdy ciąg takich znaków zaczyna się od sekwencji ESC[. ESC to znak ASCII o kodzie dziesiętnym 27, bywa ozna-

czany również przez ^[. Znak ten jest dla komputera sygnałem, że zostanie wysłany ciąg znaków sterujących. Po ESC następuje znak [(lewy nawias kwadratowy), mówiący o rozpoczęciu się ciągu znaków sterujących. To, jaki znak zostanie użyty do zakończenia ciągu znaków sterujących, zależy od funkcji realizowanej przez ten ciąg.

W poniższym opisie ANSI.SYS przyjęto następujące oznaczenia:

F1...F10 - klawisze funkcyjne od 1 do 10.

A1...A10 - klawisze funkcyjne od 1 do 10 wciśnięte razem z klawiszem Alt.

C1...C10 - klawisze funkcyjne od 1 do 10 wciśnięte razem z klawiszem Ctrl.

S1...S10 - klawisze funkcyjne od 1 do 10 wciśnięte razem z klawiszem Shift.

ANSI.SYS można wykorzystać do kontroli znaczenia klawiszy, pozycji kursora i atrybutów ekranu.

1. Znaczenie klawiszy

ESC[*p1*;*p2*p - parametr *p1* oznacza kod klawisza, któremu chcemy nadać nowe (lub przywrócić stare) znaczenie. Parametr *p2* określa nadawane klawiszowi znaczenie - może to być łańcuch znaków (ujęty w cudzysłowy "") lub/i kod ASCII.

Należy koniecznie pamiętać o tym, aby tzw. kody rozszerzone (ang. extended codes), czyli kody zdefiniowane przez ANSI .SYS, były zawsze poprzedzone znakami 0; (zero i średnik). Zero poprzedzające kod rozszerzony uprzedza system, że kod taki zostanie użyty.
Średnik pełni rolę separatora kodów.

KOD	KLAWISZ
3	- znak null
15	- Shift-Tab
16÷25	- Alt-Q, W, E, R, T, Y, U, I, O, P
30÷38	- Alt-A, S, D, F, G, H, J, K, L
44÷50	- Alt-Z, X, C, V, B, N, M
59÷68	- F1...F10
71	- Home
72	- klawisz "kursor do góry"

73 - PgUp
75 - klawisz "kursor w lewo"
77 - klawisz "kursor w prawo"
79 - End
80 - klawisz "kursor do dołu"
81 - PgDn
82 - Ins
83 - Del
84÷93 - S1...S10
94÷103 - C1...C10
104÷113 - A1...A10
114 - <Ctrl-PrtSc>
115 - <Ctrl-"kursor w lewo">
116 - <Ctrl-"kursor w prawo">
117 - <Ctrl-End>
118 - <Ctrl-PgDn>
119 - <Ctrl-Home>
120÷131 - <Alt-1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, =>
132 - <Ctrl-PgUp>

2. Pozycja kursora

ESC[*p1*;*p2*H - ustawienie kursora w określonym miejscu ekranu. *P1* może przybierać wartości od 1 do 80 (współrzędna pozioma kursora), *p2* - od 1 do 25 (współrzędna pionowa kursora). W trybie 40-znakowym *p1* może mieć wartość maksymalną 40.
ESC[*p*A - przemieszczenie kursora o *p* linii w górę,
ESC[*p*B - przemieszczenie kursora o *p* linii w dół,
ESC[*p*C - przemieszczenie kursora o *p* kolumn w prawo,
ESC[*p*D - przemieszczenie kursora o *p* kolumn w lewo,
ESC[6n - powoduje podanie przez komputer aktualnej pozycji kursora. Format odpowiedzi: ESC[*p1*;*p2*R, gdzie parametry *p1* i *p2* są odpowiednio współrzędnymi poziomą i pionową,
ESC[s - zapamiętanie bieżącej pozycji kursora,
ESC[u - przywołanie zapamiętanej pozycji kursora,
ESC[2J - wyczyszczenie ekranu i umieszczenie kursora w jego lewym górnym rogu,
ESC[k - usunięcie zawartości linii na prawo od kursora.

3. Atrybuty ekranu

ESC[*p1*;...;*pnm* - *p1*..*pn* to parametry używane do sterowania atrybutami pracy ekranu. Mogą one przybierać następujące wartości:

0 - powoduje anulowanie wszystkich uprzednio ustawionych atrybutów pracy ekranu,
1 - powoduje zwiększenie jasności świecenia znaku (wytłuszczenie),
4 - powoduje podkreślenie znaku,
5 - powoduje migotanie znaku,
7 - powoduje przedstawienie znaku w aktualnym kolorze tła na tle o aktualnym kolorze znaków (tzw. przedstawienie znaku w inwersie), przeważnie mamy do czynienia z przedstawianiem ciemnego znaku na jasnym tle,
8 - uniemożliwia wyświetlenie znaku,
30 - ustawienie koloru wyświetlanych znaków na czarny,
31 - ustawienie koloru wyświetlanych znaków na czerwony,
32 - ustawienie koloru wyświetlanych znaków na zielony,
33 - ustawienie koloru wyświetlanych znaków na żółty,
34 - ustawienie koloru wyświetlanych znaków na niebieski,
35 - ustawienie koloru wyświetlanych znaków na karmazynowy,
37 - ustawienie koloru wyświetlanych znaków na biały,
40 - ustawienie koloru tła na czarny,
41 - ustawienie koloru tła na czerwony,
42 - ustawienie koloru tła na zielony,
43 - ustawienie koloru tła na żółty,
44 - ustawienie koloru tła na niebieski,
45 - ustawienie koloru tła na karmazynowy,
46 - ustawienie koloru tła na turkusowy,
47 - ustawienie koloru tła na biały.

ESC[=*ph* - służy do ustawienia trybu pracy ekranu i może przybierać następujące wartości:

0 - tryb pracy 40x25 znaków, czarno-biały,
1 - tryb pracy 40x25 znaków, kolorowy,
2 - tryb pracy 80x25 znaków, czarno-biały,
3 - tryb pracy 80x25 znaków, kolorowy,
4 - tryb pracy 320x200 punktów, kolorowy,
5 - tryb pracy 320x200 punktów, czarno-biały,
6 - tryb pracy 640x200 punktów, czarno-biały,

7 - automatyczne przejście do następnej linii, jeżeli linia aktualna skończyła się.

ESC[=*p*1 - służy do anulowania działania ESC[=*p*h, parametry j.w. z wyjątkiem 7 (jeżeli linia skończyła się, to wszystkie znaki wprowadzane za końcem wiersza zostają pominięte).

Parametry związane z wykorzystaniem kolorów odnoszą się do karty CGA. Przykłady wykorzystania możliwości dawanych nam przez ANSI.SYS są podane w rozdziale IV.

VDISK.SYS umożliwia symulację pracy stacji dysków przy użyciu części pamięci RAM. Pewien obszar RAMu działa wtedy jak stacja dysków i można przechowywać w nim zbiory (tak samo, jak na dyskietce). W przypadku "gorącego" lub "zimnego startu" maszyny zawartość RAM-dysku (in. dysku wirtualnego) jest bezpowrotnie tracona. Utworzenie RAM-dysku powoduje zmniejszenie obszaru pamięci operacyjnej, dostępnego dla oprogramowania, o wielkość wynikającą z pojemności tego dysku. Dlatego sprzęt z niewielką ilością RAM nie daje możliwości korzystania z dysków wirtualnych o dużej pojemności. W sprzęcie o konfiguracji typu *SFDD* lub *DFDD* i pamięci np. 640 KB zainstalowanie dysku wirtualnego może być bardzo użyteczne - czas dostępu do zbiorów na takim dysku jest wielokrotnie krótszy od czasu dostępu do zbiorów zapisanych na normalnej dyskietce. Szybkość działania RAM-dysku wynika z częstotliwości pracy komputera oraz rodzaju zastosowanych układów pamięci RAM. Zatem wykorzystanie RAM-dysku może znacznie przyspieszyć naszą pracę. Polecenie DEVICE instalujące RAM-dysk ma następującą składnię:

DEVICE = [*d:*][*path*]vdisk.sys [*bbb*][*sss*][*ddd*][/e[:*m*]]

Parametr *bbb* jest używany do określenia pojemności tworzonego RAM-dysku w kilobajtach. Można podawać wartości pomiędzy 1 kB a maksymalną ilością pamięci dostępnej na określonym sprzęcie. Wartością standardową parametru *bbb* jest 64 kB. W przypadku podania nieprawidłowej wartości parametru *bbb* (mniej niż 1 kB, więcej niż cała pamięć operacyjna maszyny), zostanie on zmieniony na 64 kB. Nie jest możliwe zainstalowanie

RAM-dysku zostawiającego mniej niż 64 kB dla systemu operacyjnego.
Parametr *sss* jest używany do określenia w bajtach rozmiaru sektora na dysku wirtualnym. Może przybierać on wartości 128 (wartość standardowa), 256 i 512. Jeżeli pracujemy z małymi zbiorami, to należy dobrać mały rozmiar sektora.
Parametr *ddd* jest używany do określenia liczby pozycji (ang. directory entries) dostępnych w katalogu głównym dysku wirtualnego. Można podać od 2 do 512 pozycji (standardowo 64). Liczba ta w razie potrzeby zaokrąglana jest tak, aby na katalog użyta została całkowita ilość sektorów. Umieszczenie jednej pozycji w katalogu pochłania 32 bajty, zatem np. przy *sss*=256 i *ddd*=14 zostanie wygenerowane miejsce na 16 (a nie 14) pozycji w katalogu głównym, ponieważ 16·32=512. 512 bajtów to akurat dwa sektory, które zostaną wykorzystane na spis katalogu głównego.
Użycie parametru /e powoduje wykorzystanie dodatkowej pamięci (ang. extended memory) do zainstalowania dysku wirtualnego w komputerach typu AT.
Parametr *m* jest używany do określenia maksymalnej liczby sektorów, jaka jest używana podczas transmisji danych (standardowo 8, można podać od 1 do 8).
Jeżeli w zbiorze CONFIG.SYS umieścimy wiersz o następującej treści:

```
device=vdisk.sys 256 256 256
```

to po ponownym uruchomieniu komputera zostanie zainstalowany dysk wirtualny o pojemności 256 kB i 256-bajtowych sektorach, którego katalog główny ma miejsce na 256 pozycji.

FCBS

Polecenie FCBS ma postać

FCBS = *m*,*n*

i służy do określenia:
- całkowitej liczby zbiorów, jaka może być jednocześnie

otwarta za pomocą tzw. tablic obsługi zbiorów (ang. file control blocks, w skrócie FCB). Wartością standardową jest *m* = 4, dopuszcza się wartości *m* od 1 do 255,

- liczby zbiorów, które zostały otwarte za pomocą FCB, a które nie będą mogły być automatycznie zamknięte przez DOS w przypadku otwierania przez program aplikacyjny więcej, niż *m* zbiorów. Przed zamknięciem jest chronionych *n* pierwszych (w kolejności ich otwierania przez FCB) zbiorów.

Przez otwarcie zbioru będziemy rozumieć stworzenie możliwości przeprowadzania na nim operacji, np. pisania, czytania, zamykania itd.

FILES

Polecenie **FILES** ma postać

FILES = x

i służy do określenia maksymalnej liczby zbiorów, jaka może być jednocześnie dostępna dla systemu. Wartością standardową jest 8 (dopuszcza się wartości od 8 do 255).
Zwiększenie o 1 maksymalnej liczby zbiorów, które mogą być jednocześnie używane, powoduje zwiększenie o 39 bajtów obszaru pamięci, zajmowanego przez DOS.

LASTDRIVE

Polecenie **LASTDRIVE** ma postać

LASTDRIVE = x

i służy do określenia ostatniej ważnej nazwy stacji dysków. X może przybierać wartości od A do Z, wartością standardową jest E (odpowiada to 5 stacjom dysków, które system będzie mógł rozróżnić). Wartością minimalną jest zatem ta litera alfabetu, której numer porządkowy jest równy ilości fizycznie istniejących stacji dysków w danym komputerze.

SHELL

Polecenie SHELL ma postać

SHELL [*d:*][*path*]*nzwzbru*[.*roz*]

i służy do wskazania nowego procesora poleceń, który ma zostać wczytany zamiast COMMAND.COM.

3.10 Przetwarzanie wsadowe - zbiory typu BAT

Zbiór typu BAT jest zbiorem tekstowym zawierającym ciąg poleceń DOSu lub programów wykonywalnych pod nadzorem systemu operacyjnego. W celu wykonania przez DOS poleceń składających się na zbiór BAT należy podać polecenie

[*d:*][*path*]*nzwzbru*[.BAT] [*parametry*]

gdzie *nzwzbru* jest nazwą zbioru BAT. Spróbujmy sami utworzyć zbiór BAT:

```
A:\=>copy con: dp.bat <ENTER>
dir/p <ENTER>
<F6> <ENTER>
```

Przypominam, że <Ctrl-Z> (oznaczane jako ^Z i będące znacznikiem końca zbioru) uzyskujemy przez wciśnięcie klawisza F6. Wciśnięcie <Ctrl-Z> i potem ENTER spowoduje zapisanie zbioru DP.BAT na dyskietce w napędzie A:. Jeżeli podczas wpisywania zawartości zbioru z klawiatury zostaną wciśnięte klawisze <Ctrl-C>, to wpisywanie będzie przerwane bez zapisania zbioru na dysk.
Teraz po podaniu polecenia

```
A:\=>dp <ENTER>
```

DOS wykona polecenie zawarte w DP.BAT, czyli DIR/P.

Zajmijmy się utworzeniem nieco dłuższego zbioru BAT:

```
A:\=>copy con: prwide.bat <ENTER>
mode lpt1:132
copy %1 lpt1
mode lpt1:80
<F6> <ENTER>
```

Zbiór został nazwany PRWIDE.BAT (od ang. PRint WIDE - drukuj szeroko) i służy do wyprowadzenia na drukarkę wyszczególnionego zbioru. Przed rozpoczęciem drukowania drukarka (musi ona być zgodna programowo z IBM Graphics Printer) zostaje ustawiona w tryb 132 znaki na wiersz. Po zakończeniu drukowania ponownie jest ustawiany tryb 80 znaków na wiersz. Podanie polecenia

```
A:\=>prwide mojzbr.txt <ENTER>
```

spowoduje wyprowadzenie na drukarkę zbioru MOJZBR.TXT w trybie 132 znaki na wiersz. Nazwa zbioru, który chcemy wydrukować, jest tutaj parametrem zbioru BAT i wewnątrz tego zbioru jest reprezentowana przez zmienną %1. Każdy element linii z poleceniem uruchamiającym działanie zbioru BAT może być w tym zbiorze wykorzystany - właśnie do tego służą zmienne %0, %1, %2,...,%9. Za zmienną %0 zawsze jest podstawiana nazwa wywołanego zbioru BAT (pierwszy element linii poleceń), za %1 pierwszy parametr (drugi element linii poleceń), za %2 drugi parametr (trzeci element linii poleceń) itd. Jedynie w zbiorze AUTOEXEC.BAT zmienna %0 nie jest określona.

Jest oczywiste, że polecenia zewnętrzne zawarte w zbiorach BAT muszą być dla systemu dostępne z katalogu, z którego wykonywane są te zbiory, lub też systemowi musi być znana ścieżka dostępu do nich.

Korzystając ze zbiorów typu BAT należy przestrzegać pewnych reguł i zaleceń:

- nazwa zbioru nie może być taka sama, jak nazwa zewnętrznego lub wewnętrznego polecenia systemowego,
- w czasie realizacji przez DOS poleceń zawartych w zbiorze BAT nie należy wyjmować z napędu dyskietki, na której zbiór ten jest zapisany (polecenia są odczytywane i wykonywane sekwencyjnie - wyjęcie dyskietki powoduje wyprowadzenie

przez DOS komunikatu z prośbą o ponowne jej włożenie do napędu).

- jeżeli w zbiorze BAT następuje wywołanie innego zbioru BAT, to powrót z tego ostatniego do pierwszego może być zrealizowany jedynie poleceniem COMMAND. Przyczyną takigo stanu rzeczy jest to, że zbiory BAT są połączone są ze sobą w sposób łańcuchowy (ang. chained) - w przeciwieństwie do częściej spotykanego zanurzenia (zwanego też zagnieżdżeniem - ang. nested) jednego programu lub procedury w innym. Dlatego zaleca się, aby takie wywołanie było ostatnim poleceniem w zbiorze wywołującym.
- najszybszą realizację zbiorów BAT zapewnia uruchamianie ich z dysku wirtualnego lub twardego.
- do uruchomienia zbioru BAT wystarczy podać jego nazwę bez rozszerzenia.

W zbiorach BAT można używać specjalnych poleceń. Są one opisane poniżej.

ECHO

Polecenie wewnętrzne.

Składnia: ECHO [ON | OFF | *komunikat*]

Funkcja: wstrzymuje (ECHO OFF) i ponawia (ECHO ON) wyświetlanie nazw wykonywanych poleceń na standardowym urządzeniu wyjścia, wyprowadza na nie *komunikat*. Polecenie ECHO podane bez parametrów powoduje wyświetlenie aktualnego statusu (ON lub OFF).

FOR

Polecenie wewnętrzne.

Składnia: FOR *%%zmienna* IN (*zbiór*) DO *polecenie*

Funkcja: umożliwienie iteracyjnego wykonywania poleceń DOSu. *%%zmienna* przybiera kolejne wartości elementów (*zbioru*) i dla każdej z tych wartości jest wykonywane *polecenie*. Znaki *%%* są konieczne tylko w przypadku korzystania z polecenia FOR w zbiorze BAT - jeżeli polecenie to jest podawane za znakiem zgłoszenia się DOSu, to ma ono postać

FOR *%zmienna* IN (*zbiór*) DO *polecenie*

%zmienna może zawierać w sobie oznaczenia grup zbiorów (* i/lub ?) - wtedy są podstawiane za nią kolejno wszystkie pasujące nazwy zbiorów znajdujących się na dysku. Elementami zbioru mogą być również ścieżki dostępu.

GOTO

Polecenie wewnętrzne.

Składnia: **GOTO** *etykieta*

Funkcja: skok do linii zbioru BAT zawierającej napis *:etykieta*.

IF

Polecenie wewnętrzne.

Składnia: **IF** [**NOT**] *warunek polecenie*

Warunek może mieć następującą postać:

1. **ERRORLEVEL** *numer_błędu*
2. *łańcuch_znaków_1* == *łańcuch_znaków_2*
3. **EXIST** [*d:*][*path*]*nazwzbioru*[*.roz*]

Funkcja: warunkowe wykonanie *polecenia*. *Polecenie* jest wykonywane tylko wtedy, gdy *warunek* ma wartość logiczną 1 (true). W przeciwnym wypadku *polecenie* jest pomijane i system przechodzi do wykonania następnej linii zbioru BAT. **NOT** *warunek* jest prawdziwe wtedy, gdy *warunek* nie jest prawdziwy.

1. **ERRORLEVEL** *numer_błędu* jest prawdziwy, jeżeli poprzednio wykonany program ustawił parametr systemowy **ERRORLEVEL** na liczbę ≥ *numer_błędu*. Pomyślne zakończenie wykonywania programu ustawia **ERRORLEVEL** na 0. W przeciwnym wypadku **ERRORLEVEL** jest ustawiany na 1. Inne wartości **ERRORLEVEL** może przybierać po działaniu poleceń takich, jak np. **BACKUP** i **RESTORE**.
2. Relacja *łańcuch_znaków_1* == *łańcuch_znaków_2* jest prawdziwa, jeżeli obydwa łańcuchy są identyczne. Trzeba pamiętać o tym, że np. "A" jest różne od "a".
3. **EXIST** [*d:*][*path*]*nazwzbioru*[*.roz*] jest prawdziwe, jeżeli wyszczególniony zbiór (grupa zbiorów) istnieje.

PAUSE

Polecenie wewnętrzne.

Składnia: **PAUSE** [*komunikat*]

Funkcja: wstrzymanie wykonywania poleceń zbioru BAT aż do momentu wciśnięcia jakiegokolwiek klawisza (z wyjątkiem <Ctrl-C>) i wyprowadzenie *komunikatu*. *Komunikat* może mieć co najwyżej 121 znaków. Polecenie **PAUSE** jest używane przede wszystkim po to, aby użytkownik miał możliwość podjęcia jakichś działań (np. zmiana dyskietek) i zadecydowania o kontynuacji (lub przerwaniu) wykonywania poleceń zawartych w zbiorze BAT.

REM

Polecenie wewnętrzne.

Składnia: **REM** [*komunikat*]

Funkcja: wyprowadzenie *komunikatu*. Samo polecenie również pojawia się z *komunikatem*.

SHIFT

Polecenie wewnętrzne.

Składnia: **SHIFT**

Funkcja: "przesunięcie" wartości parametrów zbioru BAT o jeden w lewo. Jeżeli np. %0=PLIKBAT, %1=PAR1 i %2=PAR2, to po wystąpieniu polecenia **SHIFT** zmienne %i przybiorą następujące wartości: %0=PAR1, %1=PAR2, %2 będzie pusta ("%2"="").

A oto przykład prostego zbioru BAT, w którym zostały wykorzystane niektóre z omówionych powyżej poleceń (tekst zbioru należy wprowadzać bez numerów linii, które będą wykorzystane do omówienia działania poszczególnych poleceń):

```
   A:\=>copy con: kopik.bat <ENTER>
1.  echo off
2.  cls
3.  if "%1" == "" echo ********** A gdzie parametry?
4.  :pocz
5.  if "%1" == "" goto koniec
6.  echo Włóż dyskietkę z %1 do A:
7.  echo Włóż dyskietkę docelową do B:
```

```
8.  if not exist a:%1 goto niema
9.  pause Naciśnij jakiś klawisz, gdy będziesz gotowy.
10. copy a:%1 b:
11. shift
12. goto pocz
13. :niema
14. echo Nie ma %1 na A:!
15. echo Koniec pracy %0.bat - nie wszystko się udało.
16. :koniec
17. echo on
<F6>
```

Program ten służy do kopiowania wielu zbiorów (grup zbiorów). Jest uruchamiamy przez podanie jego nazwy wraz z odpowiednimi parametrami:

```
A:\=>kopik par1 par2 ... parn <ENTER>
```

Ilość parametrów nie jest ograniczona do 9 ze względu na zastosowanie polecenia SHIFT. Jedynym ograniczeniem ich liczby jest maksymalna długość linii poleceń DOSu (127 znaków). Podanie samej nazwy programu bez parametrów spowoduje wyczyszczenie ekranu i wyprowadzenie komunikatu

```
**********  A gdzie parametry?
```

Po wyprowadzeniu tego komunikatu KOPIK.BAT przywraca wyświetlanie nazw wykonywanych poleceń (tzw. włączenie echa na ekranie) i kończy działalność.
Spróbujmy teraz omówić pracę programu instrukcja po instrukcji:

1. wstrzymanie wyprowadzanie na ekran nazw poleceń wykonywanych przez system operacyjny (wyłączenie echa),
2. usunięcie wszystkich znajdujących się dotychczas na ekranie znaków i umieszczenie kursora w lewym górnym rogu ekranu (czyszczenie ekranu),
3. instrukcja warunkowa - jeżeli program został wywołany bez parametrów (relacja "%1" == "" prawdziwa), to zostaje wyprowadzony widoczny powyżej komunikat o braku parametrów.

W przypadku podania parametrów (warunek "%1" == "" nie spełniony) następuje przejście do następnej linii programu,

4. etykieta :POCZ, zostanie ona wykorzystana w dalszej części programu,
5. instrukcja warunkowa - wywołanie programu bez parametrów powoduje skok do linii z etykietą :KONIEC i wykonanie instrukcji znajdującej się bezpośrednio pod linią zawierającą etykietę. Jeżeli relacja "%1"=="" nie jest prawdziwa, następuje przejście do kolejnej linii programu,
6. wyprowadzenie komunikatu,
7. wyprowadzenie komunikatu,
8. kolejna instrukcja warunkowa sprawdza istnienie zbioru (grupy zbiorów) o nazwie, która jest aktualną wartością zmiennej %1. Jeżeli zbiór o takiej nazwie nie istnieje w aktualnym katalogu dyskietki, to wykonywany jest skok do etykiety :NIEMA (linia 14). Jeżeli zbiór istnieje, to wykonywana jest instrukcja z kolejnej linii,
9. wstrzymanie wykonywania programu aż do wciśnięcia przez użytkownika dowolnego klawisza znakowego oraz wyprowadzenie komunikatu zapisanego za poleceniem PAUSE,
10. skopiowanie zbioru (grupy zbiorów) o nazwie zgodnej z aktualną wartością zmiennej %1 z dyskietki w stacji A: na dyskietkę w stacji B:,
11. przesunięcie wartości parametrów,
12. skok do etykiety :POCZ i wykonanie instrukcji znajdującej się bezpośrednio pod linią zawierającą etykietę,
13. etykieta :NIEMA,
14. wyprowadzenie komunikatu informującego o braku na dyskietce zbioru (grupy zbiorów), którego nazwa jest aktualną wartością zmiennej %1,
15. wyprowadzenie komunikatu o kłopotach z prawidłowym wykonaniem programu,
16. etykieta :KONIEC,
17. ponowne włączenie echa.

Jak można zauważyć, polecenia DOSu wraz z poleceniami używanymi wewnątrz zbiorów BAT stanowią zorientowany problemowo język programowania. Programami w tym języku są właśnie zbiory BAT. Niekiedy mają one znaczną długość - przykłady bar-

dziej wyrafinowanego wykorzystania tych zbiorów podane w następnym rozdziale będą wymagały zapoznania się z podstawami przetwarzania tekstów. Nie jest łatwe bezbłędne wprowadzenie całego zbioru (za pomocą COPY) bezpośrednio z klawiatury, nie zachwyca też brak możliwości modyfikowania zawartości zbioru. Dlatego w przedostatniej części tego rozdziału omówimy podstawowe zlecenia wierszowego edytora tekstów EDLIN, dostarczanego razem z systemem operacyjnym.

3.11 Skrócony opis edytora EDLIN

Nazwa "edytor wierszowy" (ang. line editor) oznacza, że program przetwarza tekst wiersz po wierszu. Bezpośrednie zmiany tekstu mogą być dokonywane tylko w wierszu zwanym wierszem aktualnym i tylko w nim może znajdować się kursor. Kursor może być przemieszczany wzdłuż wiersza aktualnego, natomiast nie ma możliwości poruszania się kursorem w górę (do wierszy poprzedzających wiersz aktualny) i w dół (do wierszy następujących po wierszu aktualnym). Trzeba jasno powiedzieć, że EDLIN jest narzędziem bardzo starym i niezbyt wygodnym, szczególnie dla osób, które choć raz miały możliwość skorzystania z edytora ekranowego. Edytory ekranowe, pozwalające na wprowadzanie bezpośrednich, natychmiast w przetwarzanym tekście widocznych zmian, są bez wątpienia bardziej atrakcyjne. Dlatego użytkownicy komputerów znający programy takie, jak WordStar, WordStar 2000, ChiWriter czy edytor będący elementem kompilatora języka Turbo Pascal, mogą uśmiechać się z politowaniem, słysząc propozycję użycia EDLINa. EDLIN jednak jest dostarczany razem z systemem operacyjnym, a to oznacza, że praktycznie każdy może mieć dostęp do tego programu. Stanowi on zatem pewien standard - to zasadniczy argument za opisaniem tego właśnie edytora. Ze względu na swoją prostotę i niewykorzystywanie żadnych znaków sterujących w procesie przetwarzania tekstu, EDLIN może być całkiem użyteczny. Przypominam jednocześnie, że do edycji zbiorów tekstowych, które będą wykorzystywane bezpośrednio z poziomu systemu operacyjnego, nie należy używać edytorów posługujących się znakami sterującymi, formatującymi tekst dla potrzeb edytora. Wypro-

wadzenie na monitor tekstu zawierającego niektóre znaki sterujące (o kodzie ASCII 31 i niższym) może wywołać nieprzewidziane, a niekiedy także i przykre efekty uboczne.

Zajmijmy się teraz edytorem **EDLIN**. Nie będziemy wdawać się w jakieś szczegółowe opisy, ponieważ celem tej książki nie jest nauka przetwarzania tekstów, a nauka pracy z PC DOSem. Podane wiadomości umożliwią jednak kreowanie niewielkich zbiorów tekstowych oraz wprowadzanie w nich prostych modyfikacji, co w zupełności powinno zaspokoić nasze potrzeby. Jak już zostało powiedziane, **EDLIN** jest edytorem wierszowym. Rozpoczęcie przetwarzania tekstu następuje po podaniu polecenia w postaci

[*d:*][*path*]**EDLIN** *nzwzbru*[*.roz*]

gdzie zbiór *nzwzbru*[*.roz*] jest zbiorem tekstowym. Spróbujmy zmodyfikować nasz zbiór PRWIDE.BAT i podajmy polecenie

A:\=>edlin prwide.bat <ENTER>

Zakładamy tym samym, że systemowi znana jest ścieżka dostępu do programu **EDLIN**, a zbiór PRWIDE.BAT znajduje się w katalogu aktualnym. **EDLIN** wczyta tekst zawarty w PRWIDE.BAT i wyprowadzi komunikat

End of input file

czyli

Koniec zbioru wejściowego

Podanie nazwy nie istniejącego zbioru spowodowałoby wyprowadzenie komunikatu

New file

czyli

Nowy zbiór

Po wczytaniu zbioru przez **EDLIN** możemy przystąpić do przetwarzania tekstu. Operacje wykonywane na tekście dotyczą albo wiersza aktualnego, albo określonego przez użytkownika zakre-

su wierszy. Jeżeli prowadzimy edycję wiersza aktualnego, to można w nim używać klawiszy edycyjnych DOSu. Wszystkie wiersze tekstu są ponumerowane sekwencyjnie, a ich przenumerowanie (np. po wprowadzeniu lub usunięciu pewnej liczby wierszy) jest dokonywane automatycznie. Znakiem zgłoszenia się edytora jest gwiazdka (*), wiersz aktualny jest zaznaczany również gwiazdką. Działanie każdego polecenia EDLINa można przerwać wciskając <Ctrl-C>. Do dyspozycji mamy następujące polecenia edytora EDLIN:

a - (od ang. append) dołączenie wierszy ze zbioru na dysku do zbioru aktualnie przetwarzanego,
c - (od ang. copy) kopiowanie zawartości wierszy,
d - (od ang. delete) usunięcie określonego zakresu wierszy,
e - (od ang. end) zapisanie wprowadzonych zmian na dysku i zakończenie edycji,
i - (od ang. insert) dopisanie nowych wierszy przed wiersz bieżący,
l - (od ang. list) wyświetlenie zawartości zbioru,
m - (od ang. move) przesunięcie grupy wierszy z jednego miejsca na inne,
p - (od ang. page) wyświetlanie zawartości zbioru stronami,
q - (od ang. quit) zakończenie edycji bez zapisania wprowadzonych zmian na dysk,
r - (od ang. replace) wyszukanie w tekście i zamiana jednego łańcucha znaków na inny,
s - (od ang. search) wyszukanie łańcucha znaków w tekście,
t - (od ang. transfer) dołączenie tekstu z innego zbioru,
w - (od ang. write) zapisanie wskazanych wierszy na dysk.

Zacznijmy od wyświetlenia na ekranie zawartości zbioru PRWIDE.BAT:

```
*l <ENTER>
    1:*mode lpt1:132
    2: copy %1 lpt1
    3: mode lpt1:80
*
```

Dodajmy na początek programu kilka poleceń, które uczynią jego działanie bardziej eleganckim. Podajmy polecenie

```
*1i <ENTER>
```

które powoduje wstawienie nowego wiersza przed wiersz o numerze porządkowym 1, i napiszmy:

```
1:*echo off <ENTER>
2:*cls <ENTER>
3:*echo Czy drukarka jest przygotowana? <ENTER>
4:*pause <ENTER>
5:*<Ctrl-C>
```

Numery wierszy, dwukropki i gwiazdki są wyświetlane przez **EDLIN**. My wprowadzamy same polecenia, uzupełniające PRWIDE .BAT, kończąc każdy wiersz wciśnięciem klawisza ENTER. Powrót do zgłoszenia **EDLIN**a następuje po wciśnięciu <Ctrl-C>. Przyjrzyjmy się zmianom, jakie zaszły w tekście:

```
*l <ENTER>
    1: echo off
    2: cls
    3: echo Czy drukarka jest przygotowana?
    4: pause
    5:*mode lpt1:132
    6: copy %1 lpt1
    7: mode lpt1:80
```

Teraz spróbujmy dopisać coś na koniec tekstu:

```
*#i <ENTER>
    8*echo Gotowe! <ENTER>
    9*cls <ENTER>
   10*echo on <ENTER>
   11*<Ctrl-C>
*
```

Użyty tutaj symbol "#" (ang. hash) oznacza wiersz o numerze porządkowym o jeden większym od numeru ostatniego wiersza (natomiast jako oznaczenie wiersza aktualnego może być stosowana kropka "."). Polecenie #i powoduje zatem rozpoczęcie dopisywania nowych wierszy na koniec zbioru. Jeżeli teraz obejrzymy PRWIDE.BAT:

```
*l <ENTER>
    1: echo off
    2: cls
    3: echo Czy drukarka jest przygotowana?
    4: pause
    5: mode lpt1:132
    6: copy %1 lpt1
    7: mode lpt1:80
    8: echo Gotowe! <ENTER>
    9: cls <ENTER>
   10: echo on <ENTER>
*
```

to zorientujemy się, że popełniliśmy błąd. Wiersze 8 i 9 należy zamienić miejscami, ponieważ wyczyszczenie ekranu po wyprowadzeniu komunikatu "Gotowe!" uniemożliwiłoby przeczytanie go. Do zamiany wierszy posłuży nam polecenie m:

```
*8,8,10m <ENTER>
```

Dwa pierwsze numery wierszy określają obszar, jaki ma być przeniesiony (w tym przypadku jest to tylko jeden wiersz o numerze 8), trzecia liczba wskazuje wiersz, przed który ma się odbyć przeniesienie. Rezultat powinien być następujący:

```
*l <ENTER>
    1: echo off
    2: cls
    3: echo Czy drukarka jest przygotowana?
    4: pause
    5: mode lpt1:132
    6: copy %1 lpt1
    7: mode lpt1:80
    8: cls
    9: echo Gotowe!
   10: echo on
*
```

Ostatnim poleceniem EDLINa, które omówimy na przykładzie edycji zbioru PRWIDE.BAT, będzie polecenie r, służące do zamiany

jednego łańcucha znaków na inny. Przyjmijmy, że chcemy zmienić wiersz nr 6 z

```
copy %1 lpt1
```

na

```
copy %1 prn
```

W tym celu podajemy polecenie

```
*1,9?rlpt1<F6>prn <ENTER>
```

(Po raz ostatni przypominam, że symbol <F6> oznacza wciśnięcie klawisza funkcyjnego F6, dającego kod <Ctrl-Z>). Dwie pierwsze liczby (1 i 9) określają obszar tekstu, w którym ma nastąpić zamiana. Znak zapytania "?" oznacza, że przed każdą zamianą użytkownik zostanie poproszony o zgodę na jej dokonanie. Jeżeli znak zapytania nie zostałby podany, to akceptacja użytkownika nie byłaby wymagana. Zamieniany jest pierwszy podany łańcuch znaków (od polecenia r aż do wystąpienia <Ctrl-Z>) na drugi podany łańcuch znaków (od wystąpienia <Ctrl-Z> do <ENTER>). Ponieważ "lpt1" ma być zamienione na "prn" tylko w linii nr 6, na pytanie O.K.? o akceptację zamiany odpowiadamy kolejno N, Y, N.

EDLIN daje możliwość wprowadzenia do zbiorów znaków sterujących (o kodzie ASCII niższym od 32). Najważniejszym dla nas znakiem sterującym, którym będziemy często operować w przykładach zawartych w następnym rozdziale, jest ESC (kod 27D). Żeby uzyskać ESC w zbiorze przetwarzanym przez **EDLIN**, należy wcisnąć <Ctrl-V> i lewy nawias kwadratowy [. Podczas edycji wiersza będzie to wyglądać tak:

```
1:*^V[
```

natomiast podczas wyświetlania zawartości zbioru:

```
1:^[
```

3.12 Skrócony opis edytora pakietu SideKick

Edytory tekstów zgodne ze standardem WordStar to przede wszystkim popularne i bardzo dobre edytory firmy Borland, zawarte w programach takich, jak SideKick, Turbo Pascal, Turbo Prolog, Turbo C. Są to edytory pełnoekranowe, oferujące o wiele większe możliwości w dziedzinie przetwarzania tekstów, niż **EDLIN**.

Standard WordStar poznamy na przykładzie edytora będącego elementem pakietu SideKick. SideKick jest programem rezydentym, co umożliwia wszechstronne zastosowanie jego edytora tekstów - nie tylko do pisania krótkich zbiorów BAT, ale także np. programów w językach, których kompilatory są pozbawione własnych edytorów tekstu (liczne implementacje języka FORTRAN), lub których edytory nie nadają się do użytku (edytor dostarczany przez Ashton Tate w pakiecie dBase III+ jest co najmniej o klasę gorszy od edytora SideKicka).

Przed uruchomieniem starszych wersji SideKicka należy najpierw wczytać program Nokey, który pozwoli na skorzystanie z nielegalnej kopii SideKicka. Podajemy zatem polecenie

```
A:\=>nokey <ENTER>
```

(realizowane przez zbiór NOKEY.COM), a następnie

```
A:\=>sk <ENTER>
```

Nie zapomnijmy o tym, że SideKick musi zostać zainstalowany za pomocą programu SKINST.COM. Ma to szczególne znaczenie wtedy, gdy karta sterownika graficznego w używanym przez nas sprzęcie nie jest taka sama jak karta, na którą SideKick został wcześniej zainstalowany (np. dostaliśmy SideKicka od kolegi używającego CGA, a nasz komputer ma HGC) - po próbie uruchomienia SideKicka na monitorze nie pojawią się już żadne litery. Należy wtedy przeprowadzić gorący start (<Ctrl-Alt-Del>) i zainstalować SideKicka.
Bezpośrednio po uruchomieniu SideKicka wywołujemy główne menu (rys. 2) programu przez jednoczesne wciśnięcie klawiszy <Ctrl-Alt> (w nowszych wersjach SideKicka można za pomocą

```
C:\=>
```

```
┌ SideKick Main Menu ┐
│                    │
│ F1   Help          │
│ F2   NotePad       │
│ F3   Calculator    │
│ F4   caLendar      │
│ F5   Dialer        │
│ F6   Ascii-table   │
│ F7   Setup         │
│ Esc  exit          │
└────────────────────┘
```

```
↑↓—move bar. Select by pressing a highlighted letter, a function key, or ↵
```

Rys. 2. Menu główne pakietu SideKick

```
C:\=>
```

```
                                    Version 1.52A
Notefile
   Name:      TEMAT.MUZ
   Directory: C:\JEZYKI\TP
   Paste delay factor: 0
   Size: 45152 chars (Set with SKINST)
Appointment calendar
   Last name: APP
   Name:      APPOINT
   Directory: \
Telephone directory
   Name:      PHONE.DIR
   Directory: \
```

```
F1-help  F2-save file setup  F3-save window setup  F4-save both  ESC-exit
```

Rys. 3. Opcja Setup pakietu SideKick

programu instalacyjnego zmienić sposób jego wywołania, np. na <LeftShift-RightShift>, <Alt-LeftShift itd.>. Z menu wybieramy najpierw opcję Setup (przez wciśnięcie klawisza s lub F7). W części Notefile opcji Setup wpisujemy nazwę (Name) zbioru, który będziemy używać do prowadzenia notatek, oraz katalog (Directory), w którym zbiór ten będzie się znajdować (rys. 3). Zmiana dopuszczalnej wielkości notatnika (zbioru przeznaczonego na notatki) odbywa się za pomocą programu SKINST.COM. Po wpisaniu nazwy i katalogu w opcji Setup wciskamy klawisz F2, aby zapisać wprowadzone zmiany do zbioru SK.COM. Zmiany te zostaną zapisane, jeżeli SideKick był uruchamiany z katalogu aktualnego i katalog ten nadal jest katalogiem aktualnym. Od tej pory po każdym uruchomieniu SideKicka i wywołaniu opcji NotePad (notatnik) z głównego menu (litera n lub klawisz F2), jako notatnik będzie automatycznie wczytywany zbiór, którego nazwa została podana w opcji Setup. Wyjście z opcji Setup do systemu operacyjnego następuje po wciśnięciu klawisza Esc. Potem ponownie aktywujemy główne menu i wybie-

```
C:\=>
```

```
C:\JEZYKI\TP\TEMAT.MUZ          Line 1    Col 1   Insert    Indent
M8/1 4d , 2: 2g , 2c , 2d 2a ¦

2g , 2cis 2h , 4a 2d , 2: 2c ¦

4d , 2: 2h , 2c , 2d 2a ¦

2g , 2g 2h , 4g ¦

4c 4e 2fis , 2: 2d 2a 2e 2fis , 4c 4d 2gis , 2: 2d 2fis 2h ¦

a , gis , 2d fis a , 1: e h , 2d 2a 2c fis , 1: gis , 2a , 1: h ¦

4c 4e 2fis , 2: 2d 2a 2e 2fis , 4c 4d 2gis , 2: 2d 2fis 2h ¦

h , a , gis , g h , 4a 4d 4e 4fis ¦

2d 2g , 2g 2h , 2fis 2a , 2d 2a 2e 2c ¦

2d h , 1: a , 2g 2h c , 1: d , 2d 2a 2c 2e , 2fis 2a ¦

F1-help F2-save F3-new file F4-import data F9-expand F10-contract Esc-exit
```

Rys. 4. Okno edytora tekstowego pakietu SideKick

ramy z niego opcję NotePad (rys. 4). Na ekranie pojawia się okno, które będzie zawierać przetwarzany tekst. Wielkość i położenie okna na ekranie można regulować. Wprowadzenie programu w tryb zmiany wielkości okna uzyskujemy przez wciśnięcie klawisza F9 (powiększanie okna) lub F10 (zmniejszanie okna). Wielkość okna możemy teraz zmieniać za pomocą klawiszy sterujących kursorem (strzałki). Powrót do edycji tekstu następuje po ponownym wciśnięciu klawisza odpowiednio F9 lub F10. Zmiana położenia okna na ekranie jest możliwa po wciśnięciu klawisza ScrollLock. Klawisze sterujące kursorem powodują przesuwanie się okna po ekranie. Po umieszczeniu okna w żądanym miejscu należy ponownie wcisnąć klawisz ScrollLock, co wprowadzi program w tryb edycji tekstu. Jeżeli uregulowaliśmy wielkość i położenie okna na ekranie, to w celu zapisania tych zmian wciskamy klawisz Alt. Na ekranie (oprócz notatnika, na jego tle) pojawia się główne menu programu. Wybierany opcję Setup i wciskamy klawisz F3, co powoduje zapisanie aktualnej wielkości i położenia okien. Powrót do trybu edycji tekstu nastąpi po wciśnięciu klawisza Esc.

Wpisywanie tekstu do edytora odbywa się mniej więcej tak, jak na maszynie do pisania. Po wpisaniu wiersza tekstu przejście do następnego wiersza uzyskiwane jest poprzez wciśnięcie klawisza ENTER. Jeżeli cały ekran zostanie zapełniony tekstem, po przejściu do nowego wiersza wszystkie wiersze przesuną się o jeden do góry, a najwyższy wiersz przestanie być widoczny (oczywiście, nie jest on tracony). Na górze okna notatnika widoczna jest linia statusowa, zawierająca informację o aktualnie przotwarzanym zbiorze, oraz o wierszu i kolumnie, w której znajduje się kursor, trybie wpisywania i automatycznej tabulacji. Na dole ekranu jest wyświetlana ściągawka:

F1 - pomoc,
F2 - zapisanie przetwarzanego zbioru na dysk,
F3 - pobranie nowego zbioru do przetwarzania,
F4 - rozpoczęcie wczytania tekstu z ekranu do notatnika,
F9 - rozszerzenie okna,
F10 - zmniejszenie okna,
Esc - wyjście z edytora,

We wszystkich poniżej podanych opisach przyjęto zasadę, że (jak zawsze) oznaczenie np. <Ctrl-S> dotyczy jednoczesnego

wciśnięcia klawiszy Ctrl i S, a oznaczenie np. <Ctrl-Q><S> dotyczy wciśnięcia klawiszy najpierw jednocześnie Ctrl i Q, a potem S.

Omówmy teraz polecenia powodujące zmianę położenia kursora w przetwarzanym tekście:
- znak w lewo: <Ctrl-S> lub ←,
- znak w prawo: <Ctrl-D> lub →,
- wiersz do góry: <Ctrl-E> lub ↑,
- wiersz do dołu: <Ctrl-X> lub ↓,
- słowo w lewo: <Ctrl-A> lub <Ctrl-←>,
- słowo w prawo: <Ctrl-F> lub <Ctrl-→>,
- przesunięcie tekstu o ekran do góry: <Ctrl-R> lub PgUp,
- przesunięcie tekstu o ekran do dołu: <Ctrl-C> lub PgDn,
- początek wiersza: <Ctrl-Q><S> lub Home,
- koniec wiersza: <Ctrl-Q><D> lub End,
- początek ekranu: <Ctrl-Q><E> lub <Ctrl-Home>,
- koniec ekranu: <Ctrl-Q><X> lub <Ctrl-End>,
- początek tekstu: <Ctrl-Q><R> lub <Ctrl-PgUp>,
- koniec tekstu: <Ctrl-Q><C> lub <Ctrl-PgDn>,
- powrót na poprzednią (ostatnią) pozycję kursora: <Ctrl-Q><P>.

Tekst można wprowadzać albo w trybie dopisywania (w wierszu statusowym widoczny napis Insert), albo w trybie zamieniania (w wierszu statusowym widoczny napis Overwrite). Przełączanie między tymi trybami odbywa się przez wciśnięcie <Ctrl-V> (lub <Ins>). W trybie dopisywania znak nowo wpisywany nie zastępuje znaku już istniejącego w tekście, lecz jest dostawiany przed znak, na którym znajduje się kursor. W trybie zamieniania nowo wpisywany znak zostaje wstawiony na miejsce tego znaku, na którym znajduje się kursor.

SideKick pozwala na operacje na tzw. blokach tekstu:
- oznaczenie początku bloku: <Ctrl-K><B>,
- oznaczenie końca bloku: <Ctrl-K><K>,
- oznaczenie pojedynczego słowa jako blok: <Ctrl-K><T>,
- przełączanie między wyróżnieniem bloku w tekście a brakiem wyróżnienia: <Ctrl-K><H>,
- skopiowanie bloku: <Ctrl-K><C> (początek kopii bloku jest umieszczany na bieżącej pozycji kursora),

- przemieszczenie bloku: <Ctrl-K><V> (początek bloku jest umieszczany na bieżącej pozycji kursora),
- wczytanie bloku z dysku: <Ctrl-K><R>,
- zapisanie bloku na dysk: <Ctrl-K><W>. Blok można również przesłać na drukarkę, podając jako zbiór docelowy PRN, jest też możliwe bezpośredniego wydrukowanie bloku przez podanie polecenia <Ctrl-K><P>,
- usunięcie bloku: <Ctrl-K><Y>,
- przemieszczenie kursora na początek bloku: <Ctrl-Q><B>,
- przemieszczenie kursora na koniec bloku: <Ctrl-Q><K>,
- alfabetyczne sortowanie zawartości bloku: <Ctrl-K><S>.

Następujące polecenia dotyczą wstawiania i usuwania tekstu:
- usunięcie znaku, na którym znajduje się kursor: <Ctrl-G> (przeważnie spotyka się instalacje wykorzystujące do tego celu klawisz Del, co jest niezgodne ze standardem WordStar),
- usunięcie znaku na lewo od kursora: <Ctrl-H> (w standardzie WordStar odpowiada to klawiszowi Del, przeważnie jednak w SideKicku zamiast klawisza Del kod Ctrl-H daje klawisz BackSpace),
- usunięcie słowa na prawo od kursora: <Ctrl-T>,
- wprowadzenia nowego, pustego wiersza: <Ctrl-N> (lub <ENTER>),
- usunięcie bieżącego wiersza: <Ctrl-Y>,
- usunięcie wszystkich znaków od bieżącej pozycji kursora do końca bieżącego wiersza: <Ctrl-Q><Y>.

Dodatkowe polecenia to:
- zakończenie przetwarzania tekstu i zapisanie zmodyfikowanego zbioru na dysk: <Ctrl-K><D> lub F2,
- umieszczenie w tekście (tam, gdzie znajduje się kursor) bieżącego czasu i daty: <Ctrl-Q><T>,
- umieszczenie bieżącego czasu i daty na końcu przetwarzanego tekstu: <Ctrl-Q><O>,
- ustawienie lewego marginesu: <Ctrl-O><L>,
- ustawienie prawego marginesu: <Ctrl-O><R>,
- przełącznik trybu akceptacji pseudografiki blokowej: <Ctrl-Q><G>,
- tabulacja: <Ctrl-Q><I> lub Tab,

- unieważnienie zmian wprowadzonych w bieżącym wierszu: <Ctrl-Q><L> (działa tylko wtedy, gdy kursor nie zmienił wiersza),
- wyszukanie wskazanego łańcucha znaków: <Ctrl-Q><F>,
- opcje wyszukania wskazanego łańcucha znaków:
  B - wyszukanie łańcucha znaków wstecz od pozycji kusora,
  i - wyszukanie *i*-tego wystąpienia wskazanego łańcucha znaków,
  U - nieuwzględnianie różnic między dużymi a małymi literami,
  W - wyszukanie tylko całych słów,
- wyszukanie wskazanego łańcucha znaków i zamiana na inny łańcuch znaków: <Ctrl-Q><A>,
- opcje wyszukania i zamiany łańcuchów znaków:
  B - wyszukanie i zamiana łańcucha znaków wstecz od pozycji kursora,
  G - wyszukanie i zamiana łańcucha znaków w całym tekście, niezależnie od bieżącej pozycji kursora,
  i - wyszukanie i zamiana *i*-tego wystąpienia wskazanego łańcucha znaków,
  U - nieuwzględnianie różnic między dużymi a małymi literami,
  W - wyszukanie i zamiana tylko całych słów,
- powtórzenie ostatniej operacji wyszukania lub wyszukania i zamiany: <Ctrl-L>.

Znaki sterujące uzyskuje się przez wciśnięcie <Ctrl-P>, a potem Ctrl-Znak_Który_Chcemy_Otrzymać. Taka operacja spowoduje pojawienie się przyciemnionego (ang. low intensity) znaku na ekranie w miejscu, w którym znajdował się kursor. W przypadku ESC, wciskamy <Ctrl-P><Ctrl-[> (niektóre wersje SideKicka, np. 1.52a, wymagają wciśnięcia <Ctrl-P><Esc>). Na ekranie pojawi się przyciemniony lewy nawias kwadratowy, oznaczający właśnie ESCape.

Trzeba zauważyć, że nie wszystkie z powyżej opisanych poleceń działają w starszych wersjach SideKicka.

4. EFEKTYWNE WYKORZYSTANIE SYSTEMU OPERACYJNEGO

4.1 O co chodzi

Jak przekonaliśmy się w rozdziale 1, komputer można złożyć z poszczególnych elementów tak, aby dokładnie spełniał nasze wymagania i pasował do naszych potrzeb. Podobnie jest z systemem operacyjnym - dzięki zastosowaniu przetwarzania wsadowego (wykorzystanie zbiorów typu BAT), można zautomatyzować niemal wszystkie, często wykonywane czynności związane z obsługą komputera i zebranych danych oraz stworzyć swoje polecenia.

4.2 Podstawy

Naszym starym zwyczajem zaczniemy od czegoś bardzo prostego. Za zadanie postawmy sobie stworzenie takiego zbioru BAT, który wyświetlałby na ekranie jakiś komunikat. Za pomocą edytora EDLIN (lub polecenia COPY CON KOMUN.BAT) utwórzmy zbiór KOMUN .BAT:

```
echo To pierwszy wiersz komunikatu
echo To drugi wiersz komunikatu
echo To trzeci wiersz komunikatu
```

Teraz wypróbujmy jego działanie - podajemy polecenie

```
A:\=>komun <ENTER>
```

i na ekranie widzimy, że komputer zawsze bardzo dokładnie robi to, co mu każemy i w żaden sposób nie potrafi odgadnąć naszych zamiarów. Efekt wykonania KOMUN.BAT jest następujący:

```
echo To pierwszy wiersz komunikatu
To pierwszy wiersz komunikatu
echo To drugi wiersz komunikatu
To drugi wiersz komunikatu
echo To trzeci wiersz komunikatu
To trzeci wiersz komunikatu
```

KOMUN.BAT trzeba poprawić - dopisać **ECHO OFF** na początku, a **ECHO ON** - na końcu zbioru:

```
echo off
echo To pierwszy wiersz komunikatu
echo To drugi wiersz komunikatu
echo To trzeci wiersz komunikatu
echo on
```

Teraz KOMUN.BAT powinien działać prawidłowo:

```
A:\=>komun <ENTER>
echo off
To pierwszy wiersz komunikatu
To drugi wiersz komunikatu
To trzeci wiersz komunikatu
A:\=>
```

Wiemy, że polecenia zawarte w zbiorach BAT są wykonywane sekwencyjnie, wiersz po wierszu. Dlatego wyświetlanie długich komunikatów za pomocą ECHO nie jest efektywne. Wyprowadzenie na ekran każdego z wierszy komunikatu wymaga odczytania ze zbioru polecenia - bardziej opłacalne jest stosowanie specjalnych zbiorów tekstowych, zawierających treść komunikatu. Utwórzmy taki zbiór o nazwie KOMUN.TXT:

```
A:\=>copy con: komun.txt<ENTER>
To jest pierwszy wiersz komunikatu
```

```
To jest drugi wiersz komunikatu
To jest trzeci wiersz komunikatu
<F6>
```

Następnym krokiem jest zmiana zawartości KOMUN.BAT. Najpierw podajemy polecenie

```
A:\=>del komun.bat <ENTER>
```

a potem

```
A:\=>copy con: komun.bat <ENTER>
echo off
type komun.txt
echo on
<F6>
```

Poprawne działanie KOMUN.BAT wymaga, aby KOMUN.TXT znajdował się w tym samym, co KOMUN.BAT, katalogu. Można też podać ścieżkę dostępu do KOMUN.TXT, jeżeli chcemy trzymać komunikaty w specjalnym katalogu - np. \HELPY:

```
echo off
type a:\helpy\komun.txt
echo on
```

Po podaniu polecenia KOMUN z łatwością można zauważyć, że komunikat jest wyświetlany sprawniej i bardziej "gładko", niż poprzednio. Poprawa sprawności działania KOMUN.BAT została osiągnięta dzięki zmniejszeniu ilości poleceń realizowanych przez plik. Poprzednio wyprowadzenie trzech wierszy komunikatu wymagało wykonania trzech kolejnych poleceń. Teraz wszystkie wiersze komunikatu są wyprowadzane za pomocą jednego polecenia.

4.3 Co dalej

Zbiory BAT mogą być wykorzystywane jako własne, specjalnie przez nas zaprojektowane i dla naszych potrzeb dostowane polecenia. Takimi "osobistymi" poleceniami (i nie tylko) bę-

dziemy zajmować się w dalszej części tego rozdziału. Ponieważ potrzeby różnych użytkowników mogą być rozbieżne, nie pokusiłem się o przedstawienie PC DOSa skonfigurowanego tak, aby zadowoliło to każdego. Położyłem jedynie nacisk na przedstawienie ogólnych zasad konfigurowania systemu (w sensie programowym), popartych szczegółowymi przykładami. Na podstawie tych przykładów każdy będzie mógł stworzyć taki zbiór BAT, jaki jest mu potrzebny i przypisać takie znaczenie klawiszom funkcyjnym (i nie tylko), jakie mu się podoba.

4.4 Przykłady, czyli sztuczki i chwyty

Wyświetlanie nazw zbiorów (po podaniu polecenia DIR) odbywa się zgodnie z kolejnością, z jaką zbiory są zapisane w katalogu dyskietki. Usystematyzowanie wyświetlanej informacji uzyskujemy poprzez podawanie "szablonów" nazw zbiorów, do których pasują tylko określone nazwy i tylko one pojawiają się na monitorze (mowa o wyświetlaniu grup zbiorów). Kiedy jednak chcemy zobaczyć nazwy, daty utworzenia i wielkości wszystkich zbiorów z danego katalogu, trudno jest poradzić sobie z nadmiarem i nieuporządkowaniem informacji przekazywanej nam przez komputer. Spróbujmy zatem zwiększyć nieco możliwości systemu operacyjnego i postarajmy się utworzyć taki zbiór BAT, którego działanie pozwoliłoby na oglądanie uporządkowanej (posortowanej) zawartości katalogów. Zbiorowi realizującemu to zadanie możemy dać nazwę SWG.BAT (od Sortuj WedługG).

Przyjrzyjmy się pierwszej wersji zbioru SWG.BAT:

```
echo off
cls
if "%2"=="" goto ratunku
goto %2
:ratunku
echo Podaj parametry!
goto end
:n
dir %1 | sort
```

```
goto end
:r
dir %1 | sort/+10
goto end
:w
dir %1 | sort/+17
goto end
:d
dir %1 | sort/+24
:end
echo on
```

Jak łatwo można zauważyć, zbiór SWG.BAT wywołujemy, podając dwa parametry:

```
SWG par1 par2
```

gdzie PAR1 oznacza nazwę stacji dysków (z dyskietką, której katalog będziemy oglądać), a PAR2 sposób sortowania. Podanie polecenia w postaci

```
A:\=>swg c: n <ENTER>
```

spowoduje wyświetlenie zawartości otwartego katalogu dysku C: w kolejności alfabetycznej. Sortować można według:
- nazwy zbioru ("PAR2"="n"),
- rozszerzenia nazwy zbioru ("PAR2"="r"),
- wielkości (rozmiaru) zbioru ("PAR2"="w"),
- daty utworzenia zbioru ("PAR2"="d").

Po kilkukrotnym uruchomieniu programu można zauważyć, że nie działa on tak dobrze, jak byśmy sobie tego życzyli. Po pierwsze, jeżeli w katalogu dyskietki mieści się wiele pozycji, to trzeba wciskać <Ctrl-S>, żeby wstrzymywać wyświetlanie. Po drugie, program nie jest łatwy w użyciu (nie jest "user friendly"). Można przekonać się o tym, prosząc jakiegoś znajomego (nawet znającego się na komputerach) o podanie polecenia SWG. Zobaczymy wtedy, że komunikat

```
Podaj parametry!
```

zawiera niewiele informacji. Widać zatem, że nasz program - aby stał się bardziej elegancki - wymaga kilku poprawek. Przede wszystkim należy utworzyć tekstowy zbiór SWG.HLP, który będzie wykorzystywany przez SWG.BAT. SWG.HLP może mieć następującą treść:

```
Polecenie SWG służy do wyprowadzenia na ekran posortowanej
w określony sposób zawartości katalogu wskazanego dysku i
powinno być podane w postaci:

       SWG par1 par2

gdzie  par1 - nazwa stacji dysków
       par2 - sposób sortowania:
       1) w/g nazwy (par2=n)
       2) w/g rozszerzenia nazwy (par2=r)
       3) w/g wielkości (par2=w)
       4) w/g daty ostatniego dostępu (par2=d)

Przykład: Podanie polecenia

       swg c: d

powoduje wyświetlenie wszystkich zbiorów katalogu dysku C:
posortowanych w/g daty ostatniego dostępu.
```

Na końcu SWG.HLP radziłbym dodać kilka pustych linii, tak, aby w zbiorze były w sumie 22 linie. Będzie to mieć znaczenie nieco później. Teraz dokonajmy zmian w SWG.BAT:

```
echo off
cls
if "%2"=="" goto ratunku
goto %2
:ratunku
type swg.hlp
goto end
:n
dir %1 | sort | more
goto end
:r
dir %1 | sort/+10 | more
goto end
```

```
:w
dir %1 | sort/+17 | more
goto end
:d
dir %1 | sort/+24 | more
:end
echo on
```

Tym razem podanie polecenia (np. przez osobę nie znającą go) bez parametrów spowoduje wyświetlenie opisu prawidłowego użycia SWG.BAT i zakończenie działania programu. Opis powinien mieć 22 linie, aby nie wykraczał poza ekran i nie pozwalał na pojawienie się znaku zgłoszenia systemu bezpośrednio pod tekstem. Jeżeli polecenie SWG podamy z odpowiednimi parametrami, to wyświetlanie każdego długiego katalogu zostanie przefiltrowane przez MORE - wypełnienie się wszystkich linii na ekranie spowoduje wstrzymanie wyświetlania i wyprowadzenie komunikatu

```
---More---
```

Dalsze wyświetlanie zawartości katalogu będzie kontynuowane po wciśnięciu jakiegokolwiek klawisza znakowego.

Może trudno w to uwierzyć, ale w naszym SWG.BAT należy wprowadzić jeszcze kilka małych modyfikacji. Celem uniknięcia bałaganu na dyskietkach (a szczególnie na twardym dysku), wszystkie zbiory tekstowe o rozszerzeniu nazwy HLP powinniśmy umieścić w osobnym katalogu, np. o wspomnianej już nazwie \HELPY. W związku z tym należy zmienić szóstą linię SWG.BAT z

```
type swg.hlp
```

na

```
type a:\helpy\swg.hlp
```

Może jednak zdarzyć się, że zbiór SWG.HLP zostanie przypadkowo stracony lub zapomnimy o skopiowaniu go znajomemu. Dlatego SWG.BAT powinien reagować prawidłowo nawet w sytuacji awaryjnej. Usuńmy zatem szóstą linię:

```
type a:\helpy\swg.hlp
```

a na miejsce wprowadźmy dwie nowe linie:

```
   if not exist a:\helpy\swg.hlp echo Nie ma SWG.HLP!
ĸ                                        Podaj parametry!
   if exist a:\helpy\swg.hlp type a:\helpy\swg.hlp
```

Znak ĸ umieszczony na początku wiersza oznacza, że jest on kontynuacją wiersza poprzedniego.

Po wszystkich tych zmianach SWG.BAT powinien przedstawiać się następująco:

```
   echo off
   cls
   if "%2"=="" goto ratunku
   goto %2
   :ratunku
   if not exist a:\helpy\swg.hlp echo Nie ma SWG.HLP!
ĸ                                        Podaj parametry!
   if exist a:\helpy\swg.hlp type a:\helpy\swg.hlp
   goto end
   :n
   dir %1 | sort | more
   goto end
   :r
   dir %1 | sort/+10 | more
   goto end
   :w
   dir %1 | sort/+17 | more
   goto end
   :d
   dir %1 | sort/+24 | more
   :end
   echo on
```

PC DOS daje możliwość przeprowadzania pewnych operacji na grupach zbiorów, których nazwy zawierają podany przez użytkownika ciąg znaków, np. DIR B:*.TXT. Nie jest natomiast przewidziane wyłączanie z takich operacji zbiorów o wskazanych nazwach. Nasz przykład będzie dotyczyć zmodyfikowania polecenia DIR tak, aby były wyświetlane tylko te nazwy zbio-

rów, które nie zawierają w sobie (nazwy, nie zbiory) określonego łańcucha znaków. Za pomocą edytora **EDLIN** lub polecenia **COPY** utwórzmy zbiór DIREX.BAT (od ang. DIRectory-EXclude, czyli katalog-z-wyłączeniem):

```
echo off
cls
if "%1" == "" goto ratunku
goto ok
:ratunku
if not exist a:\helpy\direx.hlp echo
                Nie ma DIREX.HLP! Podaj par. polecenia!
if exist a:\helpy\direx.hlp type a:\helpy\direx.hlp
goto end
:ok
dir | find /v"%1" | more
:end
echo on
```

Zasada działania DIREX.BAT jest prosta - program wykorzystuje polecenia **DIR**, **FIND** i **MORE**. Najważniejsza w DIREX.BAT jest trzeci wiersz od dołu. Wyniki wykonania polecenia **DIR** są przekazywane do **FIND**. **FIND** użyte z parametrem /v ma znaleźć wszystkie te wiersze przekazane mu przez **DIR**, które zawierają w sobie łańcuch znaków wprowadzony przez użytkownika jako zmienna %1 i pominąć je przy wyświetlaniu. Wyświetlanie wyników działania DIREX.BAT jest kontrolowane przez **MORE**. Jest tutaj jednak pewien kruczek - łańcuch znaków podany jako parametr DIREX.BAT, którego wartość przybiera zmienna %1, musi być napisany dużymi literami. **FIND** odróżnia małe litery od dużych i dlatego łańcuch znaków "bulba" nie jest tym samym, co np. "BuLBa". Tę informację należy zawrzeć w zbiorze DIREX.HLP:

```
Polecenie DIREX powinno być podane w postaci

                direx ŁAŃCUCH_ZNAKÓW

Powoduje ono wyprowadzenie na ekran wszystkich nazw zbiorów nie zawierających (w nazwie) łańcucha znaków podanego
jako parametr polecenia.
```

** Uwaga!
Parametr musi być podany dużymi literami, np.:

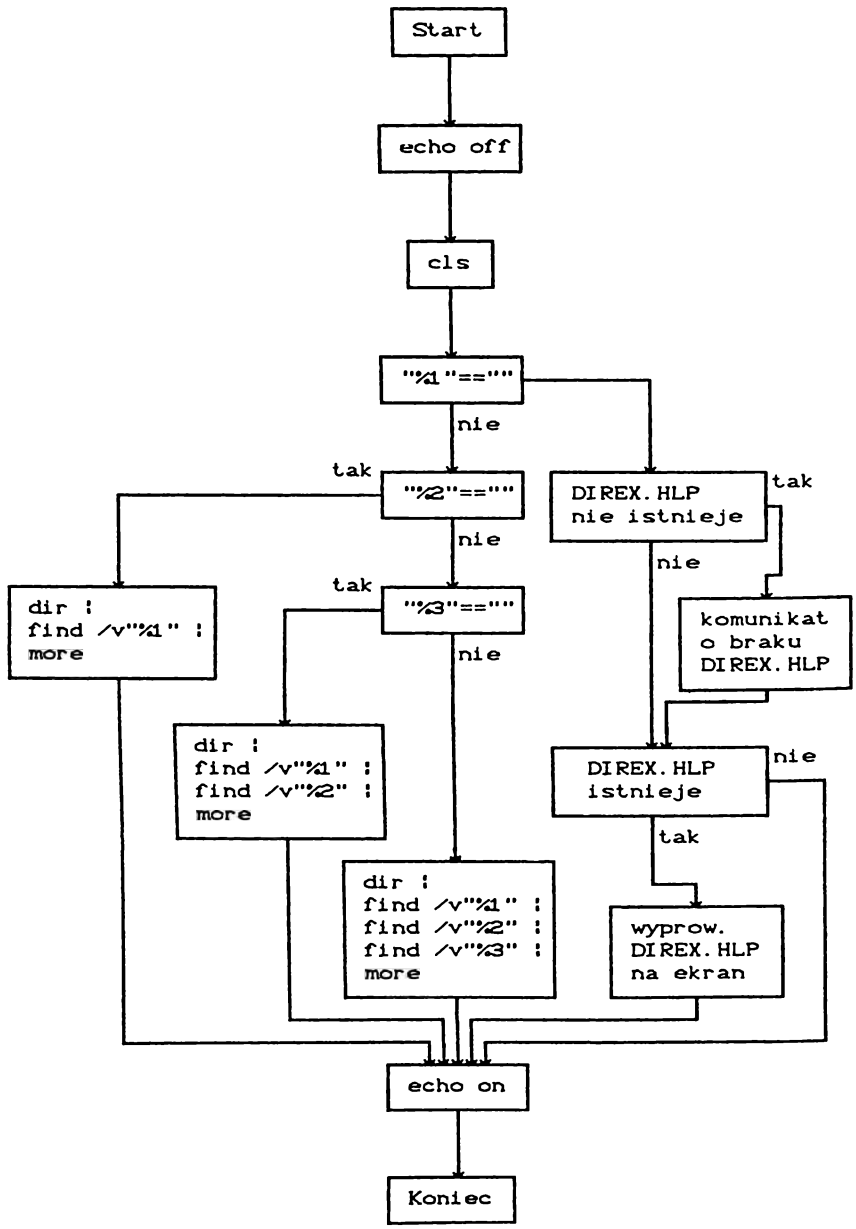
direx COM

Takie polecenie spowoduje pominięcie podczas wyświetlania wszystkich zbiorów, których nazwa zawiera łańcuch znaków "COM".

Dalszy opis działania DIREX.BAT nie jest chyba konieczny - podana powyżej treść zbioru DIREX.HLP zawiera z pewnością wystarczająco dużo informacji.
Pewnym mankamentem działania DIREX.BAT jest to, że można podać tylko jeden parametr. Zatem nie da się pominąć przy wyświetlaniu jednocześnie zbiorów,których nazwy zawierają ciągi znaków np. TXT i MOJ. Ale dla chcącego nic trudnego - łatwo jest przecież skonstruować taki zbiór DIREX.BAT, który może mieć nawet trzy parametry:

```
echo off
cls
if "%1" == "" goto ratunku
if "%2" == "" goto ok1
if "%3" == "" goto ok2
goto ok3
:ratunku
if not exist direx.hlp echo Nie ma DIREX.HLP!
                        Podaj parametry polecenia!
if exist a:\helpy\direx.hlp type a:\helpy\direx.hlp
goto end
:ok1
dir | find /v"%1" | more
goto end
:ok2
dir | find /v"%1" | find /v"%2" | more
goto end
:ok3
dir | find /v"%1" | find /v"%2" | find /v"%3" | more
:end
echo on
```

Działanie nowego DIREX.BAT opiszemy w DIREX.HLP:



Rys.5. Schemat blokowy działania zbioru DIREX.BAT

Polecenie DIREX powinno być podane w postaci

direx par1 [par2 [par3]]

Powoduje ono wyprowadzenie na ekran wszystkich nazw zbiorów nie zawierających (w nazwie) łańcuchów znaków podanych jako parametry polecenia. Można podać do 3 parametrów.

** Uwaga!

Parametry muszą być podane dużymi literami, np.

direx COM EXE

Takie polecenie spowoduje pominięcie podczas wyświetlania wszystkich zbiorów, których nazwa zawiera łańcuch znaków "COM" lub "EXE".

Każdy, kto programował choćby w BASICu, zauważy, że DIREX.BAT działa w sposób niezbyt skomplikowany. Schemat blokowy (ang. flow chart) jego pracy jest przedstawiony na rys. 5. Od razu rzuca się w oczy duża ilość "rozgałęzień" na schemacie, wynikająca przede wszystkim z istnienia specjalnej "drogi postępowania" dla każdej z dozwolonych liczb parametrów. W związku z tym kolejne usprawnienie DIREX.BAT polegać będzie na uproszczeniu algorytmu postępowania przy jednoczesnym zwiększeniu możliwości programu. Proponuję utworzyć nowy zbiór o nazwie DEXC.BAT i, wykorzystując częściowo zawartość DIREX.BAT, napisać następujący ciąg poleceń:

```
echo off
cls
if "%1" == "" goto ratunku
goto ok
:ratunku
if not exist a:\helpy\dexc.hlp echo Nie ma DEXC.HLP!
                              Podaj parametry polecenia!
if exist a:\helpy\dexc.hlp type a:\helpy\dexc.hlp
goto end
:ok
dir > dexc.tmp
:petla
find /v"%1" < dexc.tmp > dexc.tmp
```

```
shift
if "%1" == "" goto koniecpetli
goto petla
:koniecpetli
more < find /v"DEXC.TMP" < dexc.tmp
del dexc.tmp
:end
echo on
```

Schemat blokowy działania DEXC.BAT jest przedstawiony na rys. 6, natomiast jego opis znajduje się w zbiorze DEXC.HLP (poniżej):

Polecenie DEXC powinno być podane w postaci

dexc par1 [par2 ... [parn]]

Powoduje ono wyprowadzenie na ekran wszystkich nazw zbiorów nie zawierających (w nazwie) łańcuchów znaków podanych jako parametry polecenia. Długość wiersza poleceń nie może przekroczyć 127 znaków.

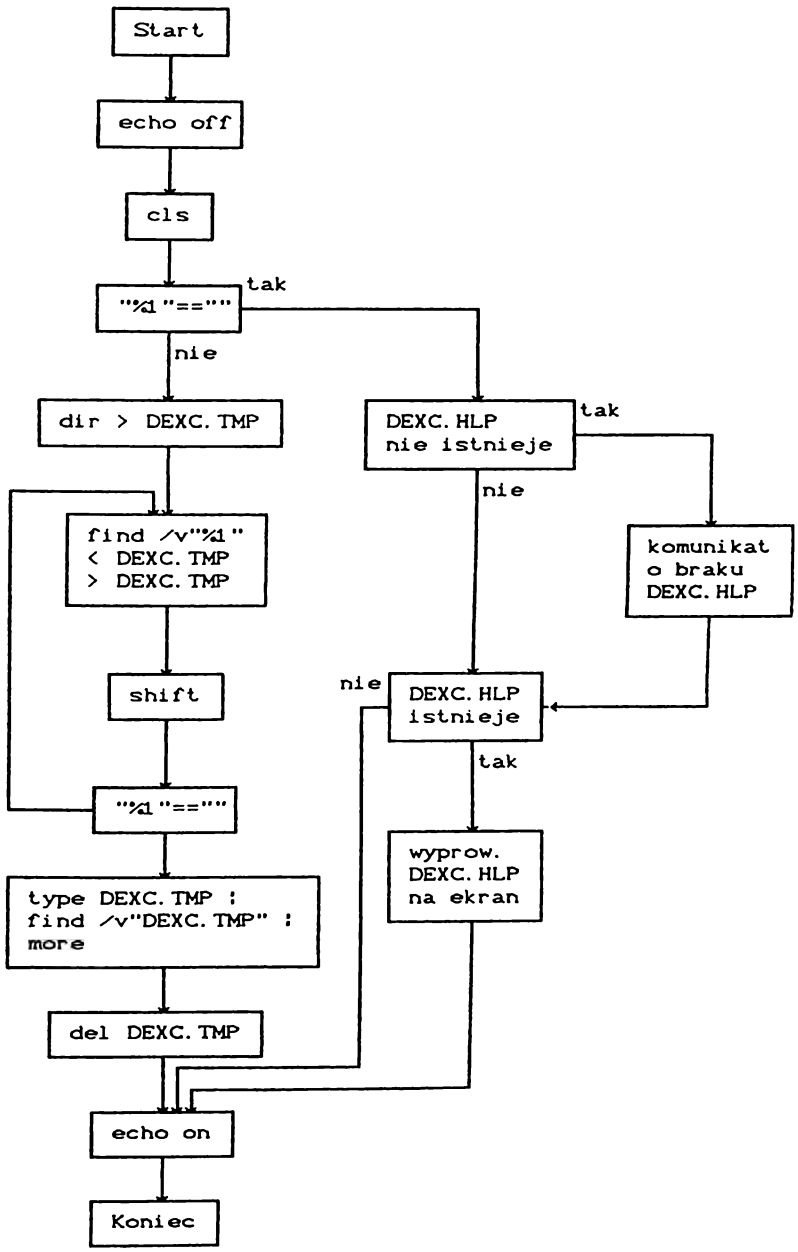
** Uwaga!

1. Podczas pracy polecenia jest tworzony czasowo na dysku zbiór DEXC.TMP, zatem polecenie nie będzie pracować z dyskietki zabezpieczonej przed zapisem.
2. Parametry polecenia muszą być podane dużymi literami, np.

dexc COM EXE

Takie polecenie spowoduje pominięcie podczas wyświetlania wszystkich zbiorów, których nazwa zawiera łańcuch znaków "COM" lub "EXE".

Mimo, że schemat blokowy i zbiór DEXC.HLP powinny stanowić zupełnie wystarczającą dla DEXC.BAT dokumentację, pozwolę sobie omówić (dla pewności, że nikt nie będzie mieć żadnych wątpliwości) działanie programu. W DEXC.BAT wykorzystaliśmy większość najatrakcyjniejszych właściwości DOSu, takich, jak strumienie, filtry, skoki w programach przetwarzania wsadowego, instrukcje warunkowe itd. Wiersze od 1 do 9 są już nam



Rys.6. Schemat blokowy działania DEXC.BAT

znane z poprzedniej wersji DEXC.BAT (DIREX.BAT) i ich rola w programie powinna być zupełnie jasna. Wiersz 10 (po etykiecie :ok) powoduje zapisanie na dysku zawartości aktualnego katalogu do zbioru DEXC.TMP. Rozszerzenie nazwy TMP pochodzi od angielskiego słowa temporary (tymczasowo, chwilowo). W 12 wierszu programu (po etykiecie :petla) następuje eliminacja tych wierszy katalogu, które zawierają łańcuch znaków podany jako pierwszy parametr polecenia DEXC. Zmiany zapisywane są ponownie w zbiorze DEXC.TMP. Kolejna wiersz programu zawiera tylko jedno polecenie **SHIFT**, które dokonuje przesunięcia wartości parametrów (patrz opis polecenia **SHIFT** w rozdziale 3). Instrukcja warunkowa w 14 wierszu sprawdza, czy istnieje jeszcze jakiś parametr, który powinien zostać uwzględniony. Jeżeli tak, to jest wykonywany skok do wiersza 11 (etykieta :pętla) i realizowana instrukcja z wiersza 12 - dla każdego z parametrów opisany powyżej ciąg instrukcji jest powtarzany. Jeżeli nie, to jest wykonywany skok do wiersza 16 (etykieta :koniecpetli) i na ekranie zostaje wyświetlona zawartość aktualnego katalogu, zmodyfikowana zgodnie z naszymi życzeniami i znajdująca się w zbiorze DEXC.TMP. Wyświetlanie odbywa się przez dwa filtry - **FIND** i **MORE**. **FIND** usuwa z wyświetlanego katalogu nazwę DEXC.TMP, działanie **MORE** jest oczywiste. Proszę zauważyć, że tym razem nie użyliśmy polecenia **TYPE**, oraz unikaliśmy potoków (znak :). Dzięki temu polecenia zawarte w DEXC.BAT będą realizowane nieco szybciej, niż w poprzedniej wersji tego programiku. Osiągnięte przez nas przyspieszenie działania DEXC.BAT wynika z tego, że DOS - jako jednozadaniowy system operacyjny - podczas przesyłania danych między programami (takimi, jak np. **TYPE** i **MORE**) za pomocą potoków musi posiłkować się tworzeniem tymczasowych zbiorów na dysku. A to zabiera czas. Dlatego zawsze wtedy, kiedy jest to możliwe, należy unikać potoków. Różnicę w szybkości działania zbiorów BAT wykorzystujących i nie wykorzystujących potoków można sprawdzić na prostym przykładzie, podanym po niniejszym opisie funkcjonowania DEXC.BAT. Ostatnie czynności wykonywane przez DEXC.BAT to usunięcie z dysku tymczasowego zbioru DEXC.TMP i wykonanie polecenia **ECHO ON**, którego działanie też jest oczywiste (jeżeli nie, to polecam ponowną lekturę końcowych fragmentów rozdziału 3).

Teraz przekonajmy się, jak to naprawdę jest z tymi potokami. Przy okazji poznamy sposób wywołania jednego zbioru BAT z innego i wykorzystanie polecenia FOR %%x IN (*zbiór*) DO *polecenie*. Oto zbiór SHOW.BAT:

```
echo off
:etykieta
for %%x in (*.%1) do command/c c:\util\typ %%x
shift
if not "%1" == "" goto etykieta
echo Wszystko zrobione.
```

W trzeciej linii zbioru SHOW.BAT następuje wywołanie nowego procesora poleceń i przesłanie mu łańcucha znaków (który jest aktualną wartością zmiennej %%x) jako parametru. Procesor ten przejmuje nadzór, interpretuje odpowiednio przesłany mu łańcuch znaków i - ze względu na parametr /c polecenia COMMAND - ponownie przekazuje nadzór zbiorowi SHOW.BAT, co pozwala na wykonanie dalszych instrukcji zawartych w SHOW.BAT. Widzimy, że nowy procesor jest wywoływany z katalogu \UTIL dysku C:. Niech będzie to zbiór TYP.BAT:

```
echo off
cls
echo Wyprowadzam na ekran %1
type %1 : more
echo.
pause
```

Omówmy teraz funkcjonowanie SHOW.BAT razem z TYP.BAT, potem zaś na chwilę zajmijmy się potokami. SHOW.BAT może być wywoływany z dowolną liczbą parametrów, które są trzyliterowymi rozszerzeniami nazw zbiorów. TYP.BAT jest używany do wyprowadzenia na ekran (przez filtr MORE) zbioru o nazwie przekazanej mu przez zmienną %%x zbioru SHOW.BAT. Zatem podanie polecenia

```
A:\=>show bat txt <ENTER>
```

spowoduje wyprowadzenie na ekran zawartości wszystkich zbiorów, które mają rozszerzenie nazwy BAT lub TXT. W czwartej

linii zbioru TYP.BAT zawartość zbioru o danej nazwie jest wyprowadzana poleceniem **TYPE**. Rezultat działania **TYPE** nie jest widoczny na ekranie, lecz jest przesyłany strumieniem na wejście polecenia **MORE**. Używamy zatem przetwarzania potokowego - a całą rzecz można zrobić i szybciej, i prościej. Wiersz

```
type %1 | more
```

należy zmienić na

```
more < %1
```

Nawet "na oko" widać, że jest to rozwiązanie bardziej rozsądne. Potwierdzają to pomiary czasu działania obu wersji pliku TYP.BAT.

Tworzenie własnych poleceń jest zatem, jak widać z powyższych przykładów, lekkie, łatwe i przyjemne. Główne ograniczenie stanowi tutaj pomysłowość (a raczej jej brak) użytkownika, no i rzeczywiste potrzeby. Można spotkać się z wyrafinowanymi w działaniu zbiorami typu BAT, które na dobrą sprawę nigdy nie są w pełni wykorzystywane, ponieważ powstały w chwili zauroczenia nowo odkrytymi możliwościami systemu i własną sprawnością programisty. Niekiedy najprostsze w działaniu zbiory BAT są najbardziej efektywne i wartościowe dla użytkownika. I tak na przykład wśród osób mających styczność z komputerami osobistymi, bardzo lubiane są jedno-, co najwyżej dwuliterowe polecenia, służące przeważnie do wywołania określonego programu z każdego katalogu na dysku (jest to szczególnie wygodne dla posiadaczy twardych dysków). Przypuśćmy, że mamy program SideKick zapisany w katalogu \SYSUT\REZYD na dysku C:, a katalogiem aktualnym jest \CHI\CHITXT. Oto polecenia, które musielibyśmy podać, gdyby nagle przyszła nam ochota na uruchomienie SideKicka (pamiętajmy, żeby przed SideKickiem uruchomić program Nokey, umożliwiający pracę pirackiej kopii SideKicka):

```
C:\CHI\CHITXT=>cd\sysut\rezyd <ENTER>
C:\SYSUT\REZYD=>nokey <ENTER>
C:\SYSUT\REZYD=>sk <ENTER>
C:\SYSUT\REZYD=>cd\chi\chitxt <ENTER>
C:\CHI\CHITXT=>
```

Nieco sprytniejszy użytkownik komputera mógłby uruchomić SideKicka w sposób następujący:

```
C:\CHI\CHITXT=>\sysut\rezyd\nokey <ENTER>
C:\CHI\CHITXT=>\sysut\rezyd\sk <ENTER>
```

W tym drugim przypadku do konieczności postukania w klawisze dochodzi konieczność ciągłego pamiętania, w którym katalogu znajduje się SideKick (dzięki poleceniu **PROMPT PQ$G** nie musimy pamiętać, który katalog jest katalogiem aktualnym). Użytkownik mało sprytny, obdarzony kiepską pamięcią i w dodatku leniwy, na pewno zrobi najpierw tak:

```
C:\=>copy con: s.bat <ENTER>
echo off
\sysut\rezyd\nokey
\sysut\rezyd\sk
cls
echo *** SideKick uruchomiony!
echo on
<F6>
```

a potem tak:

```
C:\CHI\CHITXT=>s <ENTER>
```

Od tej pory SideKick może być uruchamiany z dowolnego katalogu przez podanie jednoliterowego polecenia. W zasadzie przez jednoliterowe polecenia można wywoływać każdy często używany program. Jeżeli utworzymy kilka (czy kilkanaście) zbiorów BAT uruchamiających określone programy, to nie od rzeczy będzie sporządzenie małej ściągawki. Najpierw zapiszmy na dysku zbiór HELP.BAT:

```
echo off
cls
type a:\helpy\help.hlp
echo on
```

Zbiór HELP.HLP, używany przez HELP.BAT, może zawierać następujący komunikat:

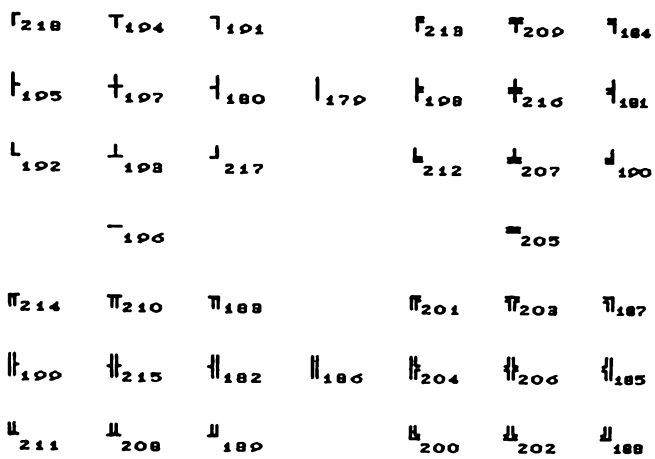
```
polecenie --> program

     b      --> dBase III+
     c      --> ChiWriter
     f      --> Fontasy
     n      --> Norton Utlities
     pr     --> Turbo Prolog
     s      --> SideKick
     tc     --> Turbo C
     tp     --> Turbo Pascal
     w      --> Wordstar 2000
```

Podanie polecenia HELP spowoduje wyprowadzenie zawartości HELP.HLP na ekran. Takie rozwiązanie jest przydatne także wtedy, kiedy z naszego sprzętu korzystają inne osoby (wskazówki dotyczące zasad tworzenia zbiorów BAT, które łatwo mogą być wykorzystywane przez innych użytkowników komputera, zostaną podane nieco później).

Należy stwierdzić, że wszystkie utworzone do tej pory zbiory tekstowe mają niezbyt ciekawą formę graficzną. Są one oczywiście bardzo przydatne i spełniają swoje zadania, ale jeżeli chodzi o wygląd, to daleko im do tego, co określa się mianem "professional look" ("wygląd profesjonalny"). Chodzi tutaj o brak ładnych ramek, migotania, wytłuszczenia czy podkreślenia tekstu. Zaraz jednak uporamy się i z tymi kłopotami. Zacznijmy od ramek. Ramki w tekście można budować za pomocą tzw. grafiki blokowej (pseudografiki). Pewnym kodom zestawu znaków ASCII są przyporządkowane symbole graficzne, które mogą być używane w trybie tekstowym tak, jak zwykłe litery. Co prawda, na końcu książki znajduje się cały zestaw znaków ASCII, ale poniżej pozwolę sobie przedstawić większość znaków (tych używanych do tworzenia ramek) grafiki blokowej w sposób uporządkowany. Obok każdego znaku został podany odpowiadający mu kod ASCII.

Informacja o istnieniu znaków graficznych i możliwości ich wykorzystania staje się użyteczna dopiero razem z informacją o tym, jak znaki te umieszczać w tekście. Każdy znak ASCII możemy "wysłać" do komputera, jeżeli wciśniemy klawisz Alt i napiszemy kod znaku za pomocą bloku numerycznego klawiatury.



Znaki grafiki blokowej

Po zwolnieniu klawisza Alt kod znaku zostanie wysłany do przetworzenia przez komputer. Tak więc, zamiast wciskać klawisz ENTER, można zastosować kombinację <Alt-13>, czyli przy wciśniętym Alt "wystukać" liczbę 13 na klawiszach bloku numerycznego (NumLock może być włączony lub wyłączony - jest to obojętne). Puszczenie klawisza Alt spowoduje wysłanie kodu ASCII 13 - co odpowiada wciśnięciu klawisza ENTER, bowiem taki właśnie jest kod powodujący powrót karetki (CR - carriage return). Zatem znakowi np. ╬ odpowiada <Alt-206>, znakowi ┌ <Alt-213> itd. Umieszczenie takich znaków w zbiorze tekstowym i ułożenie z nich ładnej, profesjonalnie wyglądającej tabelki powinno być dziecinnie łatwe. A pomyślmy tylko, jak piorunujące wrażenie może wywrzeć taka tabelka na naszych nie znających się na rzeczy znajomych. Na następnej stronie został pokazany zbiór HELP.HLP po zabiegach upiększających.

Budowanie ramki ze znaków grafiki blokowej może na początku wydawać się nieco pracochłonne - uzyskanie pojedynczego znaku wymaga kilkukrotnego wciśnięcia różnych klawiszy. Na szczęście życie może nam ułatwić nawet tak archaiczne narzędzie, jak **EDLIN**. Niech za przykład posłuży nam skonstruowanie pierwszego wiersza naszego zbioru HELP.HLP. Wiersz ten powinien wyglądać tak:

KRÓTKIE POLECENIA	
polecenie	program
b	dBase III+
c	ChiWriter
f	Fontasy
n	Norton Util.
pr	Turbo Prolog
s	SideKick
tc	Turbo C
tp	Turbo Pascal
w	Wordstar 2000

Zbiór HELP.HLP, zawierający znaki grafiki blokowej

Jeżeli zaczynamy edycję HELP.HLP, to po wywołaniu EDLINa podajmy polecenie i i napiszmy:

```
1:*      <Alt-201>aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa<Alt-187>
2:*<Ctrl-C>
```

Na ekranie będzie to widoczne jako

```
1:*      ╔aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa╗
2:*^C
```

Teraz wystarczy podać polecenie

```
*1,1ra<F6><Alt205> <ENTER>
```

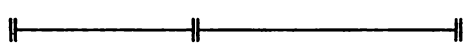
Polecenie to na ekranie będzie widoczne jako

```
*1,1ra^Z═
```

i spowoduje, że pierwszy wiersz zbioru HELP.HLP przybierze taką oto postać:

Powtarzające się elementy ramki wystarczy skonstruować raz, a potem kopiować do odpowiednich wierszy. Zauważmy, że nasza ramka składa się głównie z dwóch szablonów wiersza, a mianowicie

i



Drugi szablon można utworzyć metodą taką samą, jaka została pokazana dla pierwszego wiersza zbioru.

Po utworzeniu ramki należy wprowadzić do niej tekst - nie powinno sprawić to nikomu żadnych trudności. Jeżeli chcemy zmienić np. wiersz szósty z

na

to wystarczy podać polecenie

```
*6
6*:
6*:
```

i użyć klawiszy edycyjnych DOSu (ich działanie zostało opisane w rozdziale 3.6). Najprzydatniejszy okaże się tu niewątpliwie klawisz F1. Wciskamy go aż do momentu pojawienia się znaku ‖, a potem jeszcze czterokrotnie (uzyskujemy w ten sposób cztery spacje). Następnie wciskamy klawisz oznaczony literą b, potem znowu F1 do chwili pojawienia się znaku ‖ i dwóch spacji po nim, piszemy dBase III+ i wciskamy F3. Wprowadzanie zmian w linii 6 kończymy wciskając <ENTER>.

Jak zatem widać, EDLIN umożliwia łatwe tworzenie zbiorów tekstowych o atrakcyjnym wyglądzie. Trzeba jednak przyznać, że użytkownicy Turbo Pascala, sprawnie posługujący się operacjami na blokach, będą mieli bez porównania łatwiejsze zadanie niż osoby, których jedyną bronią jest EDLIN.

Zainstalowanie ANSI.SYS podczas uruchamiania komputera pozwala na dalsze upiększenie naszej tabelki. Nie zapomnijmy o umieszczeniu wiersza

```
device=ansi.sys
```

w zbiorze CONFIG.SYS i ponownym uruchomieniu komputera (<Ctrl-Alt-Del>), ponieważ poniżej podane przykłady wykorzystują działanie ANSI.SYS i bez zainstalowania tego sterownika nie dadzą się zrealizować.
Przypuśćmy, że w naszej tabelce chcemy wyróżnić polecenia oraz te elementy nazw programów, które odpowiadają wywołującym je poleceniom. Posłużmy się przykładem wiersza

```
|    tp      |  Turbo Pascal     |
```

Polecenie tp oraz litery T i P w nazwie Turbo Pascal powinny (po wyprowadzeniu zbioru HELP.HLP na ekran za pomocą polecenia TYPE) mieć zwiększoną intensywność świecenia:

```
|    tp      |  Turbo Pascal     |
```

Trzeba zatem przeprowadzić edycję odpowiedniego wiersza zbioru tekstowego. Według moich obliczeń powinien być to wiersz 20:

```
20*:               |    tp      |  Turbo Pascal     |
20*:
```

Wciskamy klawisze najpierw F2, a potem t (powoduje to odtworzenie zawartości wiersza 20 do pierwszego wystąpienia litery t). Następnie, po wciśnięciu klawisza Ins, wciskamy kolejno <Ctrl-V>, dwukrotnie [, potem 1 i na koniec m. Następnie dwukrotnie wciskamy F1. Powinno dać to następujący efekt:

```
20*:               |    tp      |  Turbo Pascal     |
20*:               |    ^V[[1mtp
```

Jeżeli taki mamy obraz na ekranie, to wszystko zgadza się i możemy kontynuować edycję. Jeżeli nie, to najlepiej wcisnąć <Ctrl-C> i rozpocząć edycję wiersza 20 od nowa i bardziej uważnie. Prowadząc dalej edycję (po wciśnięciu klawisza Ins), kolejno wciskamy:

<Ctrl-V> [[0 m

Następnie wciskamy F2 i <Shift-t>, a potem ponownie klawisz Ins, co pozwoli nam na wprowadzenie przed literę T sekwencji Ctrl-V[[1m. Za literę T musimy wprowadzić Ctrl-V[[0m. Z wprowadzeniem Ctrl-V[[1m przed, a Ctrl-V[[0m za literę P każdy z pewnością poradzi sobie sam. Wiersz 20, wyświetlony poleceniem l po zakończeniu edycji, powinien przedstawiać się następująco:

```
||    ^[[1mtp^[[0m      ||  ^[[1mT^[[0murbo ^[[1mP^[[0mascal    ||
```

Taką operację możemy przeprowadzić dla każdego wiersza zbioru HELP.HLP, zawierającego nazwę polecenia i programu. Jeżeli chcielibyśmy uzyskać nie tylko wytłuszczenie ciągu znaków, ale także podkreślenie, migotanie, czy (posiadacze karty CGA z kolorowym monitorem) zmianę kolorów, to na początku tego ciągu znaków należy umieścić sekwencję znaków sterujących włączającą pożądany atrybut, na końcu zaś sekwencję ESC[0m. W rezultacie podanie polecenia

```
A:\=>type \helpy\help.hlp <ENTER>
```

powinno dać mniej więcej taki wynik:

KRÓTKIE POLECENIA	
polecenie	program
b	dBase III+
c	ChiWriter
f	Fontasy
n	Norton Util.
pr	Turbo Prolog
s	SideKick
tc	Turbo C
tp	Turbo Pascal
w	Wordstar 2000

Jak pamiętamy z poprzedniego rozdziału, sekwencja ESC[*ρ*m ustawia atrybuty ekranu. W EDLINnie znak ESC (o czym powinniśmy pamiętać) jest wprowadzany właśnie jako Ctrl-V[(bezpośrednio po wprowadzeniu jest widoczny jako ^V[, potem jako ^[). Użycie parametru 1 w sekwencji sterującej powoduje większą intensywność świecenia tych znaków, które później są wyprowadzane na ekran. Sekwencja ^[[0m (czyli ESC[0m) przywraca standardowe parametry pracy ekranu. Stąd tylko te znaki, które zostały wyświetlone po ^[[1m, a przed ^[[0m, mają zwiększoną intensywność świecenia. Ta sama reguła odnosi się także do innych atrybutów pracy ekranu kontrolowanych przez ESC [*ρ*m, np. wszystkie znaki wyprowadzone między sekwencjami ESC [5m a ESC[0m będą migotać. Parametry kontrolujące pracę ekranu są podane w rozdziale 3.9 (fragment dotyczący polecenia **DEVICE** i wykorzystania ANSI.SYS). Bardzo łatwo jest przeprowadzić kilka eksperymentów, które pomogą w zrozumieniu działania sekwencji znaków sterujących. Znaki te można wysyłać nie tylko przez zapisanie ich w zbiorze tekstowym, a potem użycie polecenia **TYPE**. Pamiętamy przecież, że znak ESC można wygenerować i wysłać także za pomocą polecenia **PROMPT**. Podajmy to polecenie w postaci

```
A:\=>prompt $e[5m <ENTER>
```

Spowoduje to "zniszczenie" poprzednio zdefiniowanego znaku zgłoszenia się systemu - kursor pojawi się na początku pustej linii. Dlatego teraz należy podać polecenie

```
prompt $p$q$g <ENTER>
```

Efekt będzie nieco zaskakujący - odzyskamy nasz ulubiony znak zgłoszenia się systemu w postaci A:\=>, ale będzie on migotać. Od tej pory wszystko, co pojawi się na ekranie, będzie migotać. Jeżeli oprócz migotania chcielibyśmy uzyskać np. podkreślenie, to wystarczy podać polecenie

```
A:\=>prompt $e[4m <ENTER>
```

a potem

```
cls <ENTER>
```

Przywrócenie normalnego stanu rzeczy wymaga podania poleceń:

```
prompt $e[0m <ENTER>
prompt $p$q$g <ENTER>
```

System powinien odpowiedzieć znakiem zgłoszenia:

```
A:\=>
```

Teraz zajmiemy się rzeczami nieco bardziej użytecznymi, niż ładne ramki i teksty z mrugającymi literkami. W rozdziale 3.9 (opis polecenia **DEVICE**) zostały podane rozszerzone kody sterujące klawiszy, przyporządkowane im przez ANSI.SYS. Wykorzystując te kody, każdemu klawiszowi można nadać praktycznie dowolne znaczenie. Istnieją oczywiście pewne ograniczenia, ale o nich (i o sposobach ich obejścia) nieco później. Spróbujmy przypisać klawiszowi F7 łańcuch znaków "dir ". Na razie będziemy wykorzystywać polecenie **PROMPT**:

```
A:\=>prompt $e[0;65;"dir "p <ENTER>
prompt $p$q$g <ENTER>
```

Wciśnięcie klawisza F7 powinno dać następujący skutek:

```
A:\=>dir
```

Zatem <F7> jest teraz równoważne napisaniu "dir ". Do F8 można przypisać "type ":

```
A:\=>prompt $e[0;66;"type "p <ENTER>
prompt $p$q$g <ENTER>
```

Oprócz łańcucha znaków jest również możliwe przypisanie klawiszowi kodu znaku sterującego, np. podanie polecenia

```
A:\=>prompt $e[0;67;"dir/p";13p <ENTER>
```

spowoduje przypisanie klawiszowi F9 łańcucha znaków "dir/p" i kodu odpowiadającego wciśnięciu ENTER. Zatem wciśnięcie F9 będzie odpowiadać podaniu polecenia

```
A:\=>dir/p<ENTER>
```

Po wciśnięciu F9 polecenie DIR/P zostanie natychmiast wykonane.
Widzimy, że kody rozszerzone są poprzedzone znakami 0;. Podanie polecenia

```
A:\=>prompt $e[67;"dir/p";13p
```

spowodowałoby przypisanie łańcucha znaków "dir/p" oraz kodu klawisza ENTER literze C, która ma kod ASCII 67. Uniemożliwiłoby to używanie tej litery wprost z klawiatury. Uzyskanie C nie byłoby możliwe nawet przez wciśnięcie klawisza Alt, wprowadzenie z bloku numerycznego liczby 67 i zwolnienie Alt (efekt taki sam, jak przy bezpośrednim użyciu C). Pamiętajmy o również tym, że nadawanie nowego znaczenia klawiszom edycyjnym DOSu (F1..F6) nie jest wskazane. Poza tym, niektóre programy mogą niewłaściwie funkcjonować, jeżeli wykorzystywanym przez nie klawiszom nada się nowe znaczenie. Takim programem jest np. popularne Fontasy.

Jeżeli chcemy przypisać określonym klawiszom łańcuchy znaków odpowiadające najczęściej używanym przez nas poleceniom, to możemy uczynić to, stosując odpowiedni zbiór typu BAT. Nazwijmy go FASTKEYS.BAT:

```
prompt $e[0;104;"menu";13p
prompt $e[0;105;"cls";13p
prompt $e[0;106;"format b:/s/v";13p
prompt $e[0;107;"format "p
prompt $e[0;108;"diskcopy a: b:";13p
prompt $e[0;109;"xcopy "p
prompt $e[0;110;"cd\";13p
prompt $e[0;111;"cd ..";13p
prompt $e[0;112;"dexc "p
prompt $e[0;113;"swg "p
prompt $e[0;67;"dir/p";13p
prompt $e[0;68;"dir/w";13p
prompt $e[0;65;"dir "p
prompt $e[0;66;"type "p
prompt $p$q$g
```

Pierwszy wiersz tego zbioru przyporządkowuje polecenie MENU kombinacji klawiszy Alt-F1, czyli A1. MENU to nazwa zbioru BAT, który powodowuje wyświetlenie tekstu MENU.HLP, opisującego przyporządkowanie różnych poleceń klawiszom funkcyjnym za pomocą polecenia FASTKEYS. Utwórzmy zatem zbiory MENU.BAT:

```
echo off
cls
type a:\helpy\menu.hlp
echo on
```

i MENU.HLP:

```
                FASTKEYS  by  TomKo

     Polecenia przypisane klawiszom funkcyjnym:

       F1...F6 - klawisze edycyjne DOSu

       F7  dir                    F8  type
       F9  dir/p <J               F10 dir/w <J

       A1  menu <J                A2  cls <J
       A3  format b:/s/v <J       A4  format
       A5  diskcopy a: b: <J      A6  xcopy
       A7  cd <J                  A8  cd.. <J
       A9  dexc                   A10 swg

                     Wybieraj!
```

Znak J, który został użyty w MENU.HLP, ma kod ASCII 217. Znak < to zwykły znak mniejszości. Kombinacja <J jest bardzo często używana jako oznaczenie wciśnięcia klawisza ENTER (również w naszym zbiorze MENU.HLP).
Przypomnijmy sobie, że polecenia DEXC i SWG, przypisane klawiszom Alt-F9 (A9) i Alt-F10 (A10), są nazwami zbiorów BAT opisanych wcześniej w niniejszym rozdziale. Na wszelki wypadek przypomnę też, że polecenie CD.. zmienia katalog aktualny na katalog o poziom wyższy (patrz rozdział 3.5).
Powyżej opisanym sposobem można przyporządkować różnym klawiszom łańcuchy znaków o łącznej długości 127. Jeżeli liczba ta okaże się zbyt mała, to istnieje pewien sposób na obejście tego ograniczenia. Przypuśćmy, że nie zmieściliśmy się w 127 znakach i nie pozwala nam to na zrealizowanie przyporządkowania "DISKCOPY A: B:"<J do Alt-F5. Wiersz

```
prompt $e[0;108;"diskcopy a: b:";13p
```

zbioru FASTKEYS.BAT powinien zostać wtedy zmieniony na np.:

```
prompt $e[0;108;"d";13p
```

To przyporządkowanie będzie działać poprawnie, jeżeli utworzymy zbiór D.BAT o następującej treści:

```
diskcopy a: b:
```

Działanie klawisza A5 pozostanie wówczas takie samo, jak dla przyporządkowania mu łańcucha znaków DISKCOPY A: B:, uzyskamy natomiast oszczędność 13 znaków.
Dla osób, które mają ochotę nieco pobawić się, wykorzystując możliwości dawane przez ANSII.SYS, podamy "przepis" na upiększenie MENU.HLP. Znanymi nam już sposobami możemy wytłuścić napis "FASTKEYS by TomKo" (chociaż i tak na pewno każdy wpisze tu swoje inicjały zamiast moich), podkreślić napis "Polecenia przypisane klawiszom funkcyjnym" oraz spowodować migotanie napisu "Wybieraj!". Cały tekst ujmiemy oczywiście w eleganckie ramki:

```
╔══════════════════════════════════════════╗
╓          ^[[1mFASTKEYS  by  TomKo^[[0m                ║
╔══════════════════════════════════════════╗
╓ ^[[4mPolecenia przypisane klawiszom funkcyjnym:^[[0m ║
╔══════════════════════════════════════════╗
╓  F1...F6 - klawisze edycyjne DOSu          ║
╓                                            ║
╓  F7  dir                    F8  type       ║
╓  F9  dir/p <┘               F10 dir/w <┘   ║
╓                                            ║
╓  A1  menu <┘                A2  cls <┘     ║
╓  A3  format b:/s/v <┘       A4  format     ║
╓  A5  diskcopy a: b: <┘      A6  xcopy      ║
╓  A7  cd <┘                  A8  cd.. <┘    ║
╓  A9  dexc                   A10 swg        ║
╓                                            ║
╓              ^[[5mWybieraj!^[[0m           ║
╔══════════════════════════════════════════╗
```

Taką postać powinien mieć MENU.TXT po wyświetleniu go poleceniem **l** edytora **EDLIN**.

Rozbudowany system zbiorów BAT może być bardzo użyteczny nie tylko dla osoby tworzącej te zbiory. Jeżeli system taki zostanie skonstruowany w przemyślany sposób, to korzystać z niego mogą także inni użytkownicy komputera bez konieczności wgłębiania się w tajniki właściwego systemu operacyjnego. Uruchamianie najczęściej używanych programów powinno polegać na podawaniu prostych (jedno- lub dwuliterowych) poleceń, proste powinno być również uzyskanie informacji o sposobie uruchomienia danego programu i o aktualnym znaczeniu klawiszy funkcyjnych. Jeżeli naszym zamiarem byłoby stworzenie takiego systemu, to dla ułatwienia szybkiego konstruowania ramek można klawiszom funkcyjnym przypisać niektóre znaki grafiki blokowej. Można to zrobić za pomocą zbioru BAT i polecenia **PROMPT**:

```
prompt $e[0;84;"╔"p
prompt $e[0;85;"╗"p
prompt $e[0;86;"╚"p
prompt $e[0;87;"╝"p
prompt $e[0;88;"╦"p
prompt $e[0;89;"╩"p
prompt $e[0;90;"╠"p
prompt $e[0;91;"╣"p
prompt $e[0;92;"╬"p
prompt $e[0;93;"═"p
prompt $e[0;94;"║"p
propmt $p$q$g
```

Wykonanie takiego zbioru BAT przyporządkuje znaki graficzne kolejno klawiszom S1..S10 (Shift-F1..Shift-F10)oraz C1 (Ctrl-F1). Uzyskanie znaku grafiki blokowej, np. w EDLINie, będzie zatem znacznie prostsze i szybsze niż dotychczas. Niestety, jak już na pewno zauważyliśmy, działanie powyżej pokazanego zbioru jest bardzo powolne. Wynika to z tego, że każde polecenie **PROMPT** najpierw jest czytane z dysku, a dopiero potem wykonywane. Dla "usprawnienia naszych usprawnień" systemu operacyjnego spróbujmy posłużyć się znaną nam już metodą, wy-

korzystującą polecenie TYPE. W tym celu należy stworzyć zbiór tekstowy KEYS.ON (pamiętamy, że w EDLINie sekwencja ^[oznacza ESC i jest uzyskiwana przez wciśnięcie <Ctrl-V><[>):

```
^[[0;84;"╔"p
^[[0;85;"╗"p
^[[0;86;"╚"p
^[[0;87;"╝"p
^[[0;88;"╦"p
^[[0;89;"╩"p
^[[0;90;"╠"p
^[[0;91;"╣"p
^[[0;92;"╬"p
^[[0;93;"═"p
^[[0;94;"║"p
^[[0;104;"menu";13p
^[[0;105;"cls";13p
^[[0;106;"format b:/s/v";13p
^[[0;107;"format "p
^[[0;108;"diskcopy a: b:";13p
^[[0;109;"xcopy "p
^[[0;110;"cd;13p
^[[0;111;"cd..";13p
^[[0;112;"dexc "p
^[[0;113;"swg "p
^[[0;67;"dir/p";13p
^[[0;68;"dir/w";13p
^[[0;65;"dir "p
^[[0;66;"type "p
```

i umieścić go w katalogu \HELPY. Ten zbiór można wykorzystać do utworzenia innego zbioru tekstowego, o nazwie KEYS.OFF:

```
^[[0;84;0;84p
^[[0;85;0;85p
^[[0;86;0;86p
^[[0;87;0;87p
^[[0;88;0;88p
^[[0;89;0;89p
```

```
^[[0;90;0;90p
^[[0;91;0;91p
^[[0;92;0;92p
^[[0;93;0;93p
^[[0;94;0;94p
^[[0;104;0;104p
^[[0;105;0;105p
^[[0;106;0;106p
^[[0;107;0;107p
^[[0;108;0;108p
^[[0;109;0;109p
^[[0;110;0;110p
^[[0;111;0;111p
^[[0;112;0;112p
^[[0;113;0;113p
^[[0;67;0;67p
^[[0;68;0;68p
^[[0;65;0;65p
^[[0;66;0;66p
```

Łatwo zauważyć, że zbiór KEYS.OFF posłuży nam do przywracania pierwotnego znaczenia wcześniej przedefiniowanych klawiszy. Wykorzystanie KEYS.ON i KEYS.OFF można prześledzić na przykładzie zbioru BAT uruchamiającego program Fontasy (który jest - jak już o tym zostało wspomniane - czuły na zmienione znaczenie klawiszy funkcyjnych), zapisany w katalogu \INNE \FY. Uruchomienie Fontasy (w wersji na kartę HGC) z katalogu aktualnego polega na podaniu polecenia FYH. Nasz zbiór F.BAT pozwala na uruchomienie Fontasy z dowolnego katalogu, jednocześnie umożliwia zapomnienie o szkodliwym wpływie uprzedniego zredefiniowania klawiszy funkcyjnych na uruchamiany program:

```
echo off
cls
type a:\helpy\keys.off
cd\inne\fy
fyh
type a:\helpy\keys.on
```

```
cls
type a:\helpy\help.hlp
echo on
```

Proszę zwrócić uwagę na przedostatni wiersz zbioru F.BAT. Powoduje on - po skończeniu pracy z Fontasy - wyprowadzenie na ekran zawartości zbioru HELP.HLP,który opisuje stworzone przez nas krótkie polecenia, służące do uruchamiania określonych programów. W zasadzie każdy ze zbiorów BAT odpowiadających poleceniom wymienionym w HELP.HLP, powinien przed ostatnią instrukcją **ECHO ON** zawierać polecenie **TYPE**, powodujące wyświetlenie zawartości HELP.HLP na ekranie. Dzięki temu wszytkie osoby korzystające z naszych usprawnień będą miały ułatwione zadanie. Oczywiście, w zbiorze HELP.HLP należy dodatkowo zawrzeć informację o możliwości uzyskania dodatkowej pomocy przez wciśnięcie Alt-F1 (czyli uruchomienie MENU.BAT).

Stworzenie zintegrowanego systemu zbiorów BAT, automatyzującego nasze (i nie tylko nasze) codzienne czynności, nie powinno sprawić specjalnych kłopotów - pod warunkiem, że czytelnik dokładnie zapoznał się z przykładami podanymi w niniejszym rozdziale. Uważna analiza tych przykładów (poparta naturalnie dobrą znajomością poleceń DOSu) z pewnością ułatwi pracę z DOSem,a także pozwoli na uniknięcie przykrych niekiedy pomyłek, wynikających z "niedoczytywania" pewnych rzeczy i obdarzania komputera - zupełnie niesłusznie - intuicją.
Na zakończenie tego rozdziału przeanalizujmy funkcjonowanie przykładowych zbiorów CONFIG.SYS i AUTOEXEC.BAT. Niech zawartość CONFIG.SYS będzie następująca:

```
device = msmouse.sys
device = ansi.sys
device = vdisk.sys 64 128 64
buffers = 5
break = on
files = 12
```

Kolejne wiersz zbioru CONFIG.SYS realizują następujące czynności:

- zainstalowanie myszy Mouse Systems Mouse za pomocą sterownika dostarczanego razem z urządzeniem,
- zainstalowanie ANSI.SYS,
- utworzenie RAM-dysku o pojemności 64 kB, 128-bajtowych sektorach i katalogu z miejscem na 64 pozycje,
- zarezerwowanie miejsca na 5 buforów,
- ustawienie **BREAK** na **ON**,
- ustalenie maksymalnej liczby zbiorów, do których DOS ma jednocześnie dostęp, na 12.

A oto AUTOEXEC.BAT:

```
echo off
md d:\dos
copy command.com d:\dos
set comspec=d:\dos\command.com
path d:\dos;c:\;c:\sys;c:\sys\bajery;c:\sysut
timer/s
prompt $p$q$g
type c:\sys\bajery\keys.on
cls
echo ********** Pomoc: Alt-F1
type c:\helpy\help.hlp
echo on
```

Przez AUTOEXEC.BAT są realizowane kolejno następujące polecenia:

- wyłączenie echa,
- utworzenie katalogu \DOS na RAM-dysku D:,
- skopiowanie COMMAND.COM na RAM-dysk do katalogu \DOS,
- wskazanie systemowi, że zbiór COMMAND.COM znajduje się w katalogu \DOS RAM-dysku D: (jeżeli wystąpi konieczność ponownego wczytania COMMAND.COM, to zbiór ten zostanie wczytany z RAM-dysku - jest to szczególnie dogodne dla osób, które nie dysponują komputerem wyposażonym w twardy dysk). Po znaku równości, znajdującym się za poleceniem **COMSPEC**, nie może znajdować się spacja (byłby to niedozwolony element ścieżki dostępu do COMMAND.COM, powodujący wyprowadzenie komunikatu o błędzie),

- ustalenie ścieżki dostępu do zbiorów typu COM, EXE i BAT,
- ustawienie zegara systemowego zgodnie z zegarem czasu rzeczywistego (w komputerach typu XT),
- nadanie nowej postaci znakowi zgłoszenia się systemu,
- wyprowadzenie na ekran zbioru KEYS.ON (omówionego w niniejszym rozdziale), znajdującego się w katalogu \SYS\BAJERY na dysku C:,
- wyczyszczenie ekranu,
- wyświetlenie komunikatu "********** Pomoc: Alt-F1",
- wyprowadzenie na ekran zbioru HELP.HLP (omówionego w niniejszym rozdziale), znajdującego się w katalogu \HELPY na dysku C:,
- włączenie echa.

5. OPROGRAMOWANIE

W ostatnim już rozdziale zajmiemy się krótkim omówieniem wybranego oprogramowania, pracującego pod nadzorem systemu PC DOS. Według danych firmy Microsoft liczba dostępnych programów przekracza 40 tys. pozycji. Z tego też powodu zupełnie niemożliwe jest omówienie ich wszystkich, czy chociażby tylko tych najważniejszych. Po długim wahaniu zdecydowałem się na utrzymanie wcześniej już ukształtowanego profilu książki i poświęciłem najwięcej uwagi programom, które z założenia mają wykonywać te same czynności, co właściwy system operacyjny (tzw. nakładki na system). Tyle tylko, że dużo lepiej. Kolejno zostały omówione: Xtree, Norton Commander oraz Norton Utilities - Advanced Edition.

5.1 Xtree

Autorem programu Xtree, rozpowszechnianego przez firmę Executive Systems, Inc., jest Jeffrey C. Johnson. Program może pracować pod nadzorem systemu DOS w wersji 2.0 lub późniejszej. Na rysunku 7 widzimy ekran tytułowy programu w wersji 2.00 z roku 1985. Xtree służy przede wszystkim do pracy z twardym dyskiem - pomaga w utrzymaniu porządku, pozwala na łatwe i szybkie wykonywanie różnych operacji na katalogach, daje możliwość wyprowadzenia na drukarkę cennych informacji o organizacji i aktualnym stanie zasobów na twardym dysku. Od

programu Norton Commander (który będzie omówiony w rozdziale 5.2) Xtree różni się zorientowaniem na administrowanie twardym dyskiem, a nie na pomoc przy codziennej pracy z wieloma programami aplikacyjnymi. Świadczy o tym między innymi brak "umiejętności" ponownego wczytania się przez Xtree do pamięci operacyjnej z dysku w przypadku konfliktu z innym programem przy przydziale pamięci (Xtree nie relokuje się - niekiedy bywa to nieprzyjemne). Naturalnie z Xtree można również korzystać na sprzęcie bez twardego dysku, szczególnie przy kopiowaniu dużych ilości zbiorów (np. podczas robienia porządków na posiadanych dyskietkach - operacja taka bez wspomagania ze strony jakiegoś programu nakładkowego jest bardzo uciążliwa i czasochłonna).
W pierwszej chwili po uruchomieniu programu poleceniem

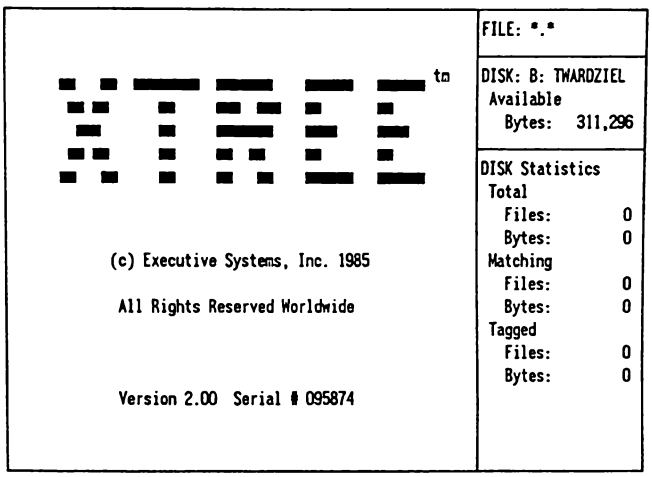
```
C:\=>xtree <ENTER>
```

jest widoczny jego ekran tytułowy (rys. 7). W tym czasie następuje zapoznanie się Xtree ze strukturą zasobów informacji zgromadzonej na aktualnym dysku (dyskietce). Potem część ekranu tytułowego zostaje zastąpiona przez ekran roboczy (rys. 8). W górnej linii jest wyświetlana aktualna ścieżka, trzy dolne linie przedstawiają skrócony opis czynności, które w danym momencie można zlecić programowi do wykonania (menu), natomiast resztę ekranu roboczego podzielimy (dla ułatwienia opisu funkcjonowania programu) na kilka obszarów, tzn.:
- okno specyfikacji nazwy,
- okno dysku,
- okno informacji o zasobach dysku,
- okno katalogów,
- okno zbiorów (małe/rozszerzone).

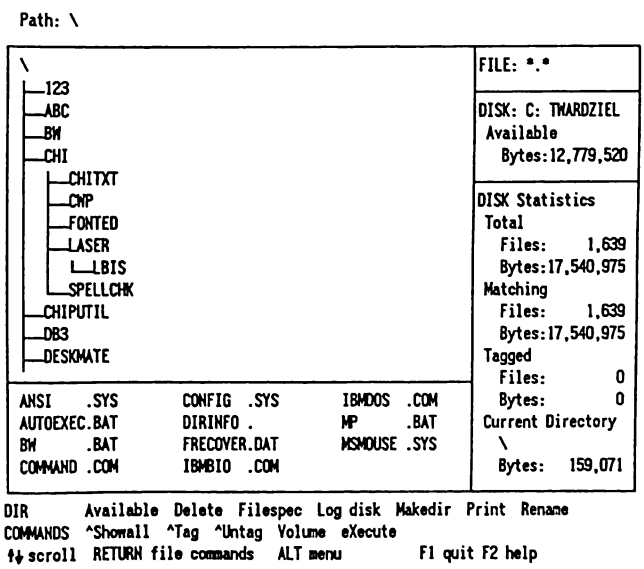
W oknie specyfikacji nazwy jest zawarta informacja o aktualnie obowiązującym wzorcu nazw zbiorów. Operacje można przeprowadzać tylko i wyłącznie na tych zbiorach, których nazwy są zgodne z wyspecyfikowanym wzorcem.

W oknie dysku jest podana nazwa dysku (volume label) i ilość wolnego miejsca (w bajtach).

W oknie informacji o zasobach na dysku (zwanym dalej krótko oknem zasobów) są podane kolejno:



Rys.7. Ekran tytułowy programu Xtree



Rys.8. Ekran roboczy programu Xtree

- liczba zbiorów na dysku,
- objętość wszystkich zbiorów na dysku,
- liczba zbiorów o nazwie zgodnej z aktualnym wzorcem,
- objętość wszystkich zbiorów o nazwie zgodnej z aktualnym wzorcem,
- liczba zbiorów zaznaczonych,
- objętość wszystkich zazanaczonych zbiorów,
- aktualny katalog,
- objętość aktualnego katalogu.

W oknie katalogów jest widoczne drzewo katalogów na aktualnym dysku. Jest również widoczny kursor - w postaci podłużnej, jasnej belki. Ponieważ kursor znajduje się w oknie katalogów, jest ono oknem aktualnym. Jako przełącznik między aktualnymi oknami (okno katalogów / okno zbiorów / rozszerzone okno zbiorów) jest wykorzystywany klawisz ENTER (RETURN).

Do zmiany katalogu aktualnego (wskazywanego w oknie katalogów przez kursor) służą klawisze sterujące kursorem: ↑, ↓, ←, →, <Home>, <End>, <PgUp> i <PgDn>.
Klawisze funkcyjne wykorzystywane przez Xtree (ich opis znajduje się zawsze w ostatniej linii na ekranie) mają następujące znaczenie:

F1 - zakończenie pracy z Xtree,
F2 - wyświetlenie tekstu pomocy,
F3 - zatrzymanie wykonywania każdego z poleceń Xtree,
F4 - włączenie / wyłączenie wyświetlania menu w trzech ostatnich liniach ekranu,
F10 - włączenie menu alternatywnego.

W oknie zbiorów są widoczne nazwy (zgodne z podanym wzorcem) zbiorów aktualnego katalogu. Przy nazwach zaznaczonych zbiorów widnieje symbol rombu.
Trzy najniższe linie ekranu zawierają polecenia związane z aktualnym oknem. Ponieważ po uruchomieniu programu jest aktualne okno katalogów, to polecenia dotyczą tego właśnie okna (Dir Commands). Działanie polecenia rozpoczyna wciśnięcie klawisza z odpowiadającą mu literą (wyróżnioną w nazwie polecenia widocznej na ekranie). Jeżeli nazwa polecenia jest poprzedzona znakiem ^, to w celu jego uruchomienia należy wcisnąć klawisz Ctrl jednocześnie z odpowiadającą poleceniu literą. Omówmy teraz po kolei działanie poszczególnych poleceń.

Available - pokazuje ilość wolnego miejsca na wskazanym dysku.

Delete - usuwa aktualny katalog (musi być on pusty).

Filespec - pozwala wprowadzić nowy wzorzec nazw zbiorów. Przypominam, że wszelkie operacje na zbiorach, wykonywane za pośrednictwem Xtree, odnoszą się tylko i wyłącznie do tych zbiorów, których nazwy są zgodne z aktualnie obowiązującym (wyświetlanym w oknie specyfikacji nazwy) wzorcem.

Log disk - zmiana aktualnego dysku.

Makedir - utworzenie nowego katalogu jako podkatalogu katalogu aktualnego.

Print - wyprowadzenie na drukarkę informacji o katalogach:

- Catalog of tagged files - wydrukowanie katalogu zaznaczonych zbiorów,
- Pathnames - wydrukowanie ścieżek dostępu,
- Tree - wydrukowanie drzewa katalogów.

Rename - zmiana nazwy aktualnego katalogu.

Show all - wyświetlenie wszystkich nazw zbiorów.

^Show all - wyświetlenie nazw wszystkich zaznaczonych zbiorów.

Tag - zaznaczenie wszystkich zbiorów w aktualnym katalogu.

^Tag - zaznaczenie wszystkich zbiorów.

Untag - usunięcie zaznaczenia wszystkich zbiorów w aktualnym katalogu.

^Untag - usunięcie zaznaczenia wszystkich zbiorów.

Volume - zmiana nazwy (etykiety) dysku.

eXecute - wykonanie programu lub polecenia DOSu. Jeżeli (po wykonaniu programu lub polecenia DOSu) kursor znajduje się bezpośrednio za znakiem zgłoszenia się systemu, to wciśnięcie ENTER powoduje powrót do Xtree. Informacja o aktualnym dysku będzie czytana ponownie.

Oprócz tych poleceń, są dostępne polecenia z tzw. alternatywnego menu okna katalogów. Menu to pojawia się na ekranie po wciśnięciu klawisza F10 lub wtedy, gdy jest wciśnięty klawisz Alt. W menu alternatywnym są dostępne następujące polecenia:

File display - przełącznik ilości informacji dodatkowej o zbiorze, wyświetlanej wraz z jego nazwą (sama nazwa zbioru / nazwa z atrybutami i wielkością zbioru / pełna informacja o zbiorze),

Sort criteria - ustalanie kolejności wyświetlania zbiorów według nazwy / rozmiaru / atrybutów / wielkości / daty i czasu ostatniej zmiany zawartości zbioru),

Tag - zaznaczenie wszystkich zbiorów, których atrybuty spełniają określone warunki,

Untag - usunięcie zaznaczenia wszystkich zbiorów, których atrybuty spełniają określone warunki,

eXecute - wykonanie programu lub polecenia DOSu. Informacja o aktualnym dysku nie jest czytana ponownie.

Po zmianie aktualngo okna na okno zbiorów (przez wciśnięcie ENTER), na dole ekranu zostanie wyświetlone menu okna zbiorów (polecenia menu okna katalogów będą dostępne dopiero po ponownym przejściu do tego okna). W menu okna zbiorów występują dwie grupy poleceń - polecenia dotyczące pojedynczych zbiorów i polecenia dotyczące zaznaczonych (grup) zbiorów. Poleceniami dotyczącymi pojedynczych zbiorów są:

Attributes - przeglądanie i/lub zmiana atrybutów aktualnego (wskazywanego przez kursor) zbioru,

Copy - skopiowanie aktualnego zbioru do innego katalogu (innego) dysku,

Delete - usunięcie aktualnego zbioru,

Filespec - wprowadzanie nowego wzorca nazwy zbiorów,

Log disk - zmiana aktualnego dysku,

Move - przesunięcie aktualnego zbioru do innego katalogu (innego) dysku,

Print - wyprowadzenie zawartości aktualnego zbioru na drukarkę,

Rename - zmiana nazwy aktualnego zbioru,

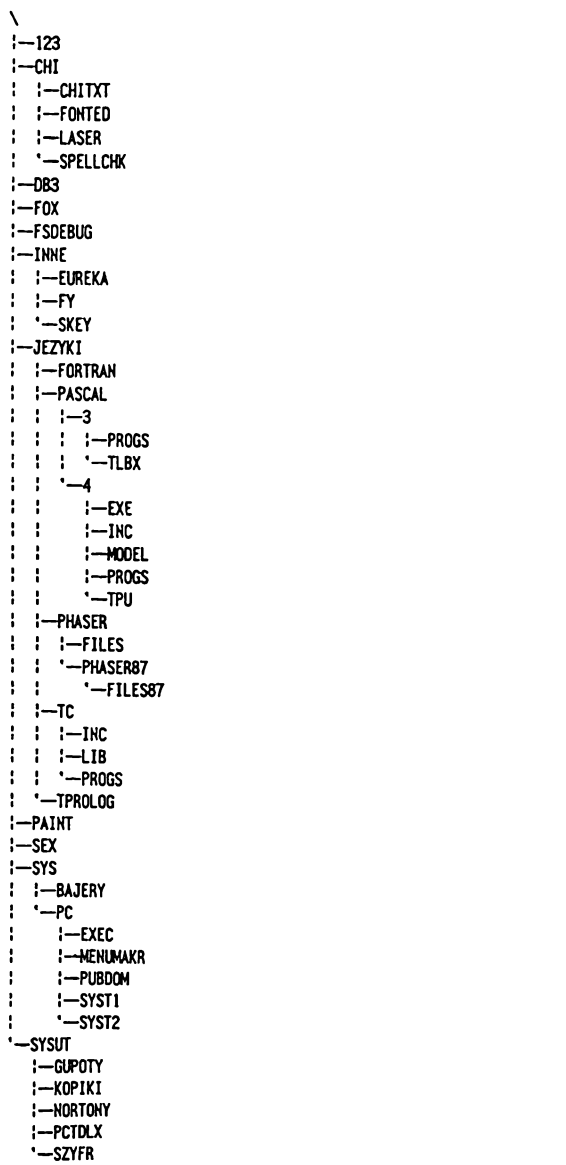
Tag - zaznaczenie aktualnego zbioru,

Untag - usunięcie zaznaczenia aktualnego zbioru,

View - przeglądanie zawartości aktualnego zbioru,

eXecute - wykonanie programu lub polecenia DOSu. Informacja o aktualnym dysku jest czytana ponownie.

Disk Volume: TWARDZIEL 6-17-88 6:05 pm Page 1



Rys.9. Obraz drzewa katalogów na twardym dysku - wynik działania programu Xtree

Disk Volume: TWARDZIEL 6-17-88 6:06 pm Page 1

\123
\CHI
\CHI\CHITXT
\CHI\FONTED
\CHI\LASER
\CHI\SPELLCHK
\DB3
\FOX
\FSDEBUG
\INNE
\INNE\EUREKA
\INNE\FY
\INNE\SKEY
\JEZYKI
\JEZYKI\FORTRAN
\JEZYKI\PASCAL
\JEZYKI\PASCAL\3
\JEZYKI\PASCAL\3\PROGS
\JEZYKI\PASCAL\3\TLBX
\JEZYKI\PASCAL\4
\JEZYKI\PASCAL\4\EXE
\JEZYKI\PASCAL\4\INC
\JEZYKI\PASCAL\4\MODEL
\JEZYKI\PASCAL\4\PROGS
\JEZYKI\PASCAL\4\TPU
\JEZYKI\PHASER
\JEZYKI\PHASER\FILES
\JEZYKI\PHASER\PHASER87
\JEZYKI\PHASER\PHASER87\FILES87
\JEZYKI\TC
\JEZYKI\TC\INC
\JEZYKI\TC\LIB
\JEZYKI\TC\PROGS
\JEZYKI\TPROLOG
\PAINT
\SEX
\SYS
\SYS\BAJERY
\SYS\PC
\SYS\PC\EXEC
\SYS\PC\MENUMAKR
\SYS\PC\PUBOOM
\SYS\PC\SYST1
\SYS\PC\SYST2
\SYSUT
\SYSUT\GUPOTY
\SYSUT\KOPIKI
\SYSUT\NORTONY
\SYSUT\PCTOLX
\SYSUT\SZYFR

Rys.10. Spis wszystkich ścieżek dostępu na twardym dysku - wynik działania programu Xtree

Poleceniami dotyczącymi zaznaczonych (grup) zbiorów są:

^Attributes - modyfikacja atrybutów zaznaczonych zbiorów,
^Copy - skopiowanie zaznaczonych zbiorów do innego katalogu (innego) dysku,
^Delete - usunięcie zaznaczonych zbiorów,
^Move - przesunięcie zaznaczonych zbiorów do innego katalogu (innego) dysku,
^Print - wyprowadzenie zawartości zaznaczonych zbiorów na drukarkę,
^Rename - zmiana nazw zaznaczonych zbiorów,
^Tag - zaznaczenie wszystkich zbiorów w aktualnym katalogu.
^Untag - usunięcie zaznaczenia wszystkich zbiorów w aktualnym katalogu.

Alternatywne menu okna zbiorów jest dostępne po wciśnięciu klawisza F10 lub wtedy, gdy jest wciśnięty klawisz Alt. To menu zawiera następujące polecenia:

Copy - skopiowanie wszystkich zaznaczonych zbiorów na inny dysk. Każdy z zaznaczonych zbiorów jest kopiowany na dysk docelowy do katalogu o tej samej nazwie, co katalog na dysku źródłowym. W razie potrzeby katalog taki jest tworzony na dysku docelowym,
File display - przełącznik ilości informacji dodatkowej o zbiorze, wyświetlanej wraz z jego nazwą (sama nazwa zbioru / nazwa z atrybutami i wielkością zbioru / pełna informacja o zbiorze),
Sort criteria - ustalanie kolejności wyświetlania zbiorów (według nazwy / rozmiaru / atrybutów / wielkości / daty i czasu ostatniej zmiany zawartości zbioru),
Tag - zaznaczenie wszystkich zbiorów, których atrybuty spełniają określone warunki,
Untag - usunięcie zaznaczenia wszystkich zbiorów, których atrybuty spełniają określone warunki,
eXecute - wykonanie programu lub polecenia DOSu. Informacja o aktualnym dysku nie jest czytana ponownie.

Jak widać, Xtree nie jest programem bardzo skomplikowanym w użyciu. Oczywiście, należy przyzwyczaić się do pewnych jego osobliwości, jak np. brak możliwości relokacji kodu (co praktycznie uniemożliwia szersze korzystanie z polecenia eXecute) czy konieczność częstej aktualizacji informacji o dysku. Kiedy jednak nauczymy się wykorzystywać wszystkie możliwości programu, z pewnością trudno będzie nam znaleźć lepszego pomocnika przy administrowaniu naszymi zasobami, zgromadzonymi na twardym dysku i licznych dyskietkach. Muszę przyznać się, że jestem gorącym zwolennikiem (i użytkownikiem) Xtree. Dlatego też, żeby zachęcić czytelnika do korzystania z tego programu, pozwoliłem sobie przedstawić rezultaty pracy Xtree na rysunkach 9 (drzewo katalogów) i 10 (wszystkie ścieżki dostępu, występujące na dysku).

5.2 Norton Commander

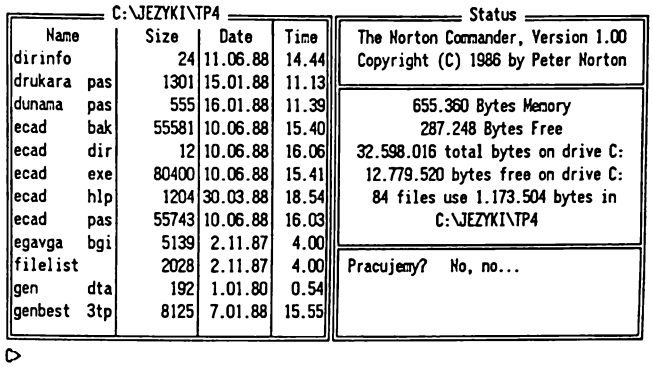
Norton Commander jest rozpowszechniany jako produkt Petera Nortona (Peter Norton Computing Inc.), mimo że został napisany przez Johna Sochę. Program ten ma pośredniczyć między użytkownikiem a systemem operacyjnym i w ten sposób ułatwiać i uprzyjemniać obsługę systemu. Omawiana wersja 1.0 z 1986 roku wymaga DOSu 2.0 lub późniejszego i co najmniej 128 KB pamięci operacyjnej. Norton Commander może pracować w dwóch wersjach, zajmujących różną ilość pamięci operacyjnej. Wersja pełna jest uruchamiana poleceniem

```
A:\=>nc <ENTER>
```

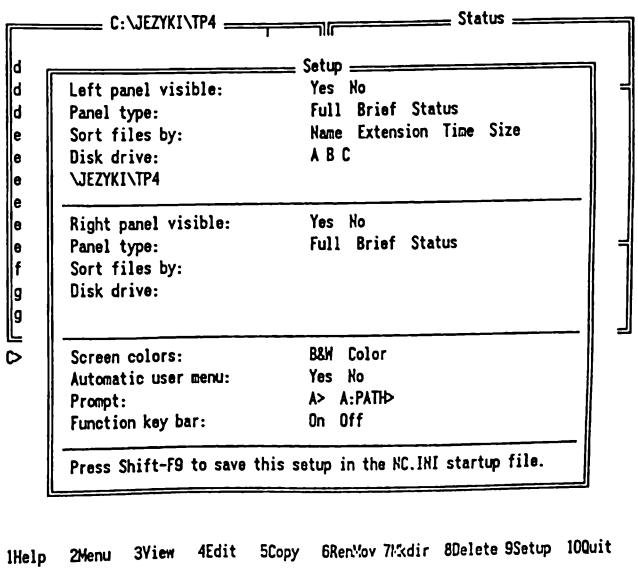
co rozpoczyna działanie zbioru NC.EXE. Tak uruchomiony Norton Commander wszystkie swoje opcje ma zawarte bezpośrednio w pamięci operacyjnej, co powoduje, że zużywa on na swoje potrzeby niemal 90 KB RAM. Wersja "mała" jest uruchamiana poleceniem

```
A:\=>ncsmall <ENTER>
```

co rozpoczyna pracę zbioru NCSMALL.EXE. Zbiór ten (zajmujący tylko 11 KB RAM) jest wspomagany zbiorem NC.EXE, który dla



Rys.11. Program Norton Commander w standardowej kofiguracji, bezpośrednio po uruchomieniu



Rys.12. Okno konfiguracji programu Norton Commannder

prawidłowego działania całości programu musi być dostępny na dysku. Zbiory NC.EXE oraz NCSMALL.EXE mogą dodatkowo współpracować ze zbiorami NC.INI i NC.MNU. NC.INI zawiera parametry dotyczące konfiguracji uruchamianego Norton Commandera, natomiast NC.MNU (tworzenie i wykorzystanie tego zbioru zostanie omówione później) zawiera menu stworzone przez użytkownika na własne potrzeby. Trzeba pamiętać o tym, że NC.INI i NC.MNU powinny być zapisane w tym samym katalogu, w którym znajdują się NC.EXE i NCSMALL.EXE. Jeżeli nie dopilnujemy tego, to zbiór NC.INI nie zostanie zauważony przez program główny przy konfigurowaniu się, a potem będziemy mieć kłopoty z korzystaniem z menu zdefiniowanego w NC.MNU. Bezpośrednio po uruchomieniu program (o ile w NC.INI nie jest zapisana inna konfiguracja) przedstawia się tak, jak na rys. 11. Na ekranie są widoczne dwa okna - z lewej strony okno aktualnego dysku, z prawej strony okno statusu. W oknie dysku są widoczne nazwy zbiorów, uporządkowane alfabetycznie (jedna z nazw - nazwa zbioru aktualnego - jest wyróżniona przez kursor w postaci podłużnej belki). Również są wyświetlane nazwy ukrytych zbiorów i katalogów. Nazwy te są zaznaczone w oknie przez symbol ▒. W oknie statusu znajdują się dane dotyczące dysku i systemu. Są tam podane informacje o ilości zaistalowanej i dostępnej pamięci operacyjnej, o całkowitej pojemności i wolnym miejscu pozostającym na dysku, o liczbie i wielkości wszystkich zbiorów w aktualnym katalogu. Na samym dole okna statusu jest wyświetlana treść zbioru tekstowego o nazwie DIRINFO. W zbiorze tym (kreowanym samodzielnie przez użytkownika) jest zawarta informacja o aktualnym katalogu lub jakakolwiek inna informacja tekstowa. Bezpośrednio pod oknami znajduje się znak zgłoszenia systemu, po którym można wprowadzać polecenia systemowe. Jeżeli takie polecenie ma zostać wykonane, to Norton Commander musi mieć dostęp do zbioru COMMAND.COM (w razie potrzeby zostanie wyprowadzony komunikat o konieczności udostępnienia tego zbioru). Najniższa linia ekranu przedstawia aktualne znaczenie klawiszy funkcyjnych:

F1 - wyprowadzenie na ekran okna z tekstem pomocy,

F2 - wyprowadzenie na ekran okna zawierającego menu stworzone przez użytkownika,

F3 - przeglądanie zawartości aktualnego zbioru,

F4 - prowadzenie edycji aktualnego zbioru (maksymalna wielkość zbioru: 20 kB),
F5 - skopiowanie aktualnego zbioru / gupy zaznaczonych zbiorów,
F6 - zmiana nazwy / przemieszczenie aktualnego zbioru / grupy zaznaczonych zbiorów,
F7 - utworzenie podkatalogu katalogu aktualnego,
F8 - usunięcie aktualnego zbioru / grupy zaznaczonych zbiorów,
F9 - wprowadzenie zmian w konfiguracji programu,
F10 - zakończenie pracy z programem.

Jeżeli uprzednio zmienialiśmy znaczenie klawiszy funkcyjnych, to Norton Commander ponownie je zredefiniuje. Można też wykorzysytwać klawisze S3...S9 (Shift-F3...Shift-F9). Mają one takie samo znaczenie, jak F3...F9 (S9 powoduje zapisanie aktualnej konfiguracji pracy programu do zbioru NC.INI), ale dotyczą zawsze pojedynczego zbioru, na którym ostanio wykonywane były jakieś operacje, lub zbioru o podanej nazwie.
Przejdźmy teraz do omówienia konfiguracji pracy programu. Jak już zostało powiedziane, konfigurację można zmieniać po wciśnięciu klawisza F9 (rys. 12). Okno konfiguracji (ang. setup window) jest podzielone na 4 poziomy. Pierwszy i drugi poziom określają atrybuty odpowiednio lewego i prawego okna:
- okno widoczne (panel visible): tak / nie (yes / no),
- typ okna (panel type): pełne / uproszczone / statusowe (full / brief / status),
- sposób sortowania zbiorów (sort files by): według nazwy / rozszerzenia / czasu / rozmiaru (name / ext./ time / size),
- obsługiwana stacja dysków,
- aktualny katalog.

Trzeci poziom okna konfiguracji służy do określenia:
- trybu pracy (screen colors): monochromatyczny / kolorowy (B&W / color),
- automatycznego wyświetlania menu użytkownika (automatic user menu): tak / nie (yes / no),
- znaku zgłoszenia się systemu (prompt): bez podawania ścieżki (A>) / ze ścieżką (A:PATH>),
- wyświetlania linii przypominającej znaczenie klawiszy fun-

kcyjnych (function key bar): linia widoczna / niewidoczna (on / off).

Czwarty poziom okna konfiguracji zawiera komunikat, informujący użytkownika o możliwości zapisania aktualnej konfiguracji w NC.INI za pomocą <Shift-F9>.

Wszystkie parametry pracy programu są wybierane w oknie za pomocą klawiszy sterujących kursorem (↑, →, ↓, ←). Wciśnięcie ENTER powoduje akceptację ustawionego parametru, natomiast użycie klawisza ESCape unieważnia zmiany wprowadzone w oknie konfiguracji.

Podczas pracy z Norton Commanderem można (jak już wiemy) podawać normalne zlecenia systemowe, które zostaną wykonane, jeżeli jest dostępny zbiór COMMAND.COM na dysku aktualnym lub na dysku w tej stacji, z której nastąpiło wczytanie się systemu. Porównując pracę z Norton Commanderem i bez niego, od razu daje się odczuć, że nakładka często potrafi oszczędzić nam wiele pisania. Dotyczy to zwłaszcza wprowadzania nazw zbiorów jako parametrów poleceń systemowych. Jeżeli jakaś nazwa zbioru zostanie wskazana kursorem (przez umieszczenie kursora na niej za pomocą klawiszy sterujących), to wciśnięcie <Ctrl-ENTER> spowoduje przepisanie tej nazwy do wiersza poleceń. Zresztą kursora w oknie dysku wcale nie trzeba przemieszczać po nazwach zbiorów za pomocą klawiszy sterujących. Wciśnięcie klawisza Alt z jakąś literą (np. <Alt-A>) spowoduje umieszczenie kursora od razu na pierwszej nazwie zbioru, zaczynającej się od danej litery (np. ABACAB.TXT). Jeżeli wciśniemy kolejną literę (a klawisz Alt będziemy dalej trzymać wciśnięty - będzie to zatem odpowiadać kombinacji np. <Alt-A-K>), to kursor zostanie umieszczony na pierwszej nazwie, zaczynającej się od dwóch danych liter (np. AKUKU.BZZ). Przypuśćmy, że w katalogu znajdują się zbiory AAA.TXT, ABC .TXT i ZZZ.TXT. Naszym zamiarem jest wysłanie ABC .TXT na drukarkę. Ponieważ Norton Commander nie współpracuje z drukarką (i jest to chyba najpoważniejsza wada tego programu), użyjemy systemowego polecenia **COPY**. Piszemy zatem za znakiem zgłosznia się systemu

```
A>copy
```

(za "copy" należy wstawić jedną spację), wciskamy najpierw <Alt-A-B> (trzymając Alt wciśnięty, wciskamy kolejno klawisze z literami a i b), a potem <Ctrl-ENTER>:

```
A>copy abc.txt
```

i dopisujemy PRN:

```
A>copy abc.txt prn
```

Taka operacja powinna wysłać ABC.TXT na drukarkę.

Jeżeli w oknie dysku kursor znajdzie się na nazwie zbioru z rozszerzeniem COM, EXE lub BAT, to wciśnięcie ENTER spowoduje uruchomienie danego programu. Działania, które powinny być podejmowane przez program w zależności od rozszerzenia nazwy zbioru można zdefiniować w zbiorze NC.EXT.

Do poruszania się po liście zbiorów w oknie dysku.służą klawisze <Home>, <End>, <PgUp>, <PgDn>, ↑, →, ↓ i ←. A oto kilka dalszych poleceń, sterujących pracą Norton Commandera:

<Ctrl-PgUp> - odpowiada CD.., można też wcisnąć ENTER, jeżeli kursor w oknie dysku znajduje się na wejściu do katalogu nadrzędnego,

<Ctrl-PgDn> - odpowiada CD *nazwa*, można też wcisnąć ENTER, jeżeli kursor w oknie dysku znajduje się na nazwie odpowiedniego podkatalogu,

<Ctrl-\> - odpowiada CD\,

<Ctrl-R> - ponowne przeczytanie zawartości aktualnego katalogu (aktualizacja informacji o katalogu),

szary <+> - zaznaczenie grupy zbiorów o nazwach zgodnych z podanym wzorcem. Przez określenie "szary <+>" (ang. grey <+>) jest rozumiany klawisz znajdujący się w bloku numerycznym,

szary <-> - usuwanie zaznaczenia grupy zbiorów o nazwach zgodnych z podanym wzorcem,

<Ins> - przełącznik zaznaczenia pojedynczego zbioru,

<Ctrl-E> - poprzednie polecenie systemowe ze stosu. Przy niewidocznych oknach (<Ctrl-O>) poprzednie polecenie systemowe ze stosu można przywoływać używając klawisza ↑,

<Ctrl-X> - następne polecenie systemowe ze stosu. Przy niewidocznych oknach następne polecenie systemowe ze stosu można przywoływać używając klawisza ↓.

<Ctrl-ENTER> - przepisanie do wiersza poleceń nazwy zbioru (katalogu), na której znajduje się kursor w oknie dysku.

<Ctrl-I> - zmiana aktywnego okna (tę samą funkcję spełnia klawisz <Tab>),

<Ctrl-O> - przełącznik wyświetlania okien,

<Ctrl-P> - przełącznik wyświetlania nieaktywnego okna,

<Ctrl-U> - zamiana okien miejscami,

<Ctrl-L> - przełącznik okno statusu / okno dysku,

<Ctrl-B> - przełącznik wyświetlania linii, opisującej znaczenie klawiszy funkcyjnych,

Dobrze jest uświadomić sobie, że przy usuwaniu zaznaczenia grupy zbiorów można podać inny wzorzec nazwy zbioru, niż obowiązujący przy zaznaczaniu. Pozwala to na bardzo efektywne wyselekcjonowanie tych zbiorów, które mają być przedmiotem różnych operacji.

W edytorze tekstu (wywołanie: F4) Norton Commandera użytkownik ma do dyspozycji następujące polecenia:

↑, →, ↓, ← - poruszanie się po tekście co jeden znak,

<Ctrl-←> - przemieszczenie kursora o słowo w lewo,

<Ctrl-→> - przemieszczenie kursora o słowo w prawo,

<PgUp>, <PgDn> - przesunięcie tekstu o ekran w górę / w dół,

<Home>, <End> - początek / koniec bieżącego wiersza,

<Ctrl-Home> - początek tekstu,

<Ctrl-End> - koniec tekstu,

<BackSpace> - usunięcie znaku na lewo od kursora,

<Del> - usunięcie znaku, na którym znajduje się kursor,

<Ctrl-T> - usunięcie słowa na prawo od kursora,

<Ctrl-W> - usunięcie słowa na lewo od kursora (tak samo działa <Ctrl-BackSpace>),

<Ctrl-Y> - usunięcie bieżącego wiersza,

<Ctrl-K> - usunięcie tekstu między kursorem a końcem bieżącego wiersza,

F1 - wyświetlenie tekstu pomocy,

F2 - zapisanie przetwarzanego zbioru na dysk,

F7 - szukanie wyspecyfikowanego łańcucha znaków w zbiorze (również przy przeglądaniu zbioru - polecenie View, uruchamiane klawiszem F7),

F10 lub Esc - zakończenie edycji.

Jeżeli nie zapiszemy na dysk zbioru po wprowadzeniu w nim zmian, to przed wyjściem z edytora (F10 lub Esc) na ekranie pojawi się okno z komunikatem o tym, że zmiany wprowadzone w zbiorze nie zostały zapisane na dysk. Możemy wtedy nową wersję zbioru zapisać (opcja "save"), lub nie (opcja "don't save"), możemy też wrócić do edycji zawartości zbioru (opcja "continue editing"). Opcje z menu w Norton Commanderze są wybierane zawsze za pomocą klawiszy sterujących kursorem i akceptowane przez wciśnięcie ENTER. Każda opcja może być wybrana również przez wciśnięcie litery, odpowiadającej pierwszej literze jej nazwy. Jeżeli chcemy przywrócić stan sprzed pojawienia się okna z propozycjami podjęcia działań przez program, to w każdej sytuacji wystarczy użyć klawisza Esc.

Edytor tekstu zawarty w Norton Commanderze można wykorzystać do utworzenia zbioru NC.MNU. Wciskamy Shift-F4, wpisujemy nazwę NC.MNU i potwierdzamy naszą chęć stworzenia nowego zbioru klawiszem ENTER. Gdy po wykonaniu tych operacji znajdziemy się już w edytorze, wpiszemy następujący tekst:

```
B: dBase III+
cd\db3
dbase
C: ChiWriter
cd\cw
cw
A: ACAD
cd\acad
acad
F: Fontasy
type \helpy\keys.off
cd\inne\fy
fyh
type \helpy\keys.on
cd\
```

Wciskamy F2 (zapisanie zbioru na dysk), F10 (lub Esc) i ponownie F2. Na ekranie pojawi się okno, z utworzonym przed chwilą przez nas menu. Opcje w nim zawarte są wybierane za pomocą klawiszy sterujących kursorem oraz klawisza ENTER, lub za pomocą litery definiującej daną pozycję w menu. I tak np. program dBase III+ z katalogu \DB3 można uruchomić, wciskając klawisz oznaczony literą b.

5.3 Norton Utilities - Advanced Edition

Peter Norton znany jest chyba wszystkim użytkownikom komputerów osobistych zgodnych z IBM. Popularne są książki tego autora, jeszcze popularniejsze programy użytkowe, pozwalające dokonywać różnych "cudów", jak np. odratowanie danych z twardego dysku po sformatowaniu. Norton Utilities - Advanced Edition (nazywane dalej w skrócie NU-AE) to ostatnia wersja (z maja 1987r.) szeroko znanego zestawu programów użytkowych Nortona. Programy te pozwalają na realizację wielu czynności, normalnie wykonywanych przez system operacyjny, w możliwie najprostszy sposób. Jednocześnie użytkownik NU-AE ma do dyspozycji bardzo "silne" narzędzie, przydatne przy intensywnej pracy z systemem i występujących wtedy komplikacjach. Oprócz dublowania czynności systemowych, NU-AE umożliwia m.in. tworzenie konwersacyjnych zbiorów BAT, generowanie dźwięku z poziomu systemu, porządkowanie katalogów, ratowanie skasowanych zbiorów, łatwe sterowanie parametrami pracy ekranu, itd. Podstawowy zestaw NU-AE (bez kilku zbiorów służących do demonstracji możliwości pakietu) składa się z 24 zbiorów, z których każdy może być wykorzystany oddzielnie. Do skonsolidowania tak dużej ilości programów w jedną całość służy tzw. Norton Integrator, (zbiór NI.EXE) uruchamiany poleceniem NI. Jest on wyposażony jest w bardzo rozbudowany system tekstów pomocy, przypominający znaczenie i sposób użycia poszczególnych elementów pakietu. Zanim przejdziemy do opisu funkcjonowania NI, zapamiętajmy, że pracę z programem kończymy przez wciśnięcie klawisza Esc lub F10. Po uruchomieniu NI ekran zostaje podzielony na trzy części. Z lewej strony ekranu jest widoczna lista dostępnych poleceń, z prawej zaś - opis aktualnego polecenia. Na dole ekranu znajduje się wiersz poleceń, z wpi-

saną nazwą aktualnego polecenia. Za nazwą można dopisać parametry polecenia. Zmiana aktualnego polecenia za pomocą klawiszy sterujących kursorem trwa dosyć długo (ze względu na konieczność wyświetlenia nowego tekstu pomocy dla każdego nowego polecenia). NI umożliwia szybkie wyszukanie potrzebnego polecenia (ang. speed search) - w tryb szybkiego wyszukania program jest wprowadzany przez wciśnięcie klawisza Tab. Wiersz poleceń zostaje wtedy wymazany. Wyszukanie nowego aktualnego polecenia następuje poprzez wciskanie kolejno klawiszy z literami odpowiadającymi literom nazwy polecenia. Wciśnięcie spacji powoduje opuszczenie trybu szybkiego wyszukania i umożliwia wpisanie parametrów polecenia w wierszu poleceń. Na przykład, jeżeli chcemy podać polecenie FA /R (zmieniające atrybuty zbioru, ang. file attributes), to wciskamy kolejno Tab, literę f i spację, po czym dopisujemy parametr /R. Jeżeli naszym zamiarem byłoby podanie polecenia FR /SAVE (zapisanie na dysku dodatkowej informacji, pozwalającej na późniejsze odratowanie danych po przypadkowym sformatowaniu dyku), to po wciśnięciu Tab musimy wcisnąć f, r, spację i dopisać /SAVE. Co prawda każdego z programów wchodzących w skład NU-AE można używać oddzielnie, ale NI jest wielce pomocny, gdy chodzi o podpowiedzenie sposobu wykorzystania danego polecenia, podania parametrów itp. Dlatego osoby, które nie mają zamiaru włączyć pakietu Nortona do grupy programów stale używanych, powinny korzystać z pomocy i podpowiedzi zawartych w NI. Bardzo ułatwia to pracę - skorzystanie z NU-AE nie wymaga uczenia się na pamięć nowego, rozbudowanego zestawu poleceń.

Dokładnemu omówieniu wszystkich programów, wchodzących w skład pakietu Nortona, należałoby poświęcić osobny rozdział - szczególnie dotyczy to głównego programu NU-AE, tzn. NU.EXE. Dlatego też bliżej przyjrzymy się tylko kilku programom.

ASK

Zbiór ASK.EXE można wykorzystywać do tworzenia interakcyjnych zbiorów wsadowych. Polecenie ASK jest podawane w postaci

```
ASK "komunikat", [lista-klawiszy]
```

i powoduje najpierw wyświetlenie komunikatu, a potem oczekuje na wciśnięcie jednego z klawiszy z podanej listy. W zależności od tego, który klawisz został wciśnięty, ASK ustawia parametr systemowy **ERRORLEVEL** na odpowiedni poziom. Wciśnięcie pierwszego klawisza z listy powoduje ustawienie **ERRORLEVEL** na 1, drugiego na 2 itd. Prześledźmy działanie ASK na prostym przykładzie. Utwórzmy zbiór ABC.BAT:

```
echo off
ask "Wciśnij a, b lub c: ", abc
if errorlevel 3 goto c:
if errorlevel 2 goto b:
echo Mam cię! Wcisnąłeś a.
goto koniec
:b
echo Mam cię! Wcisnąłeś b.
goto koniec
:c
echo Mam cię! Wcisnąłeś c.
:koniec
echo on
```

Funkjonowanie ABC.BAT nie wymaga chyba omawiania.

DT

Zbiór DT.EXE służy do testowania dysków. Polecenie DT należy podawać w następującej postaci:

DT [*d:*] [*nzwzbru*[*.roz*]] [*/parametry*]

Parametry mogą być następujące:

/b - przeprowadzenie jednocześnie testu dysku i poprawności zapisu zbiorów,
/cn - oznaczenie klastra *n* jako uszkodzonego,
/d - testowanie całego dysku,
/f - testowanie tylko poprawności zapisu zbiorów,
/log - przygotowanie do wyprowadzenia wyników działania DT na drukarkę lub do zbioru,

/m - przepisanie zagrożonych klastrów w bezpieczne miejsce na dysku,

/s - przeprowadzenie testu także podkatalogów.

Po podaniu polecenia DT bez parametrów na ekranie pojawia się komunikat:

```
Select DISK test, FILE test, or BOTH
Press D, F, or B ...
```

Wciśnięcie klawisza d spowoduje przeprowadzenie testu całego obszaru dysku, f - poprawności zapisu zbiorów na dysku, b - obu testów.

FR

Zbiór FR.EXE usuwa skutki przeprowadzonego formatowania twardego dysku. Polecenie FR jest podawane w następującej postaci:

```
FR [d:][/save]
```

Użycie parametru /save powoduje zapisanie na dysku danych, wykorzystywanych potem przy odratowywaniu dysku po sformatowaniu. Na wszelki przypadek dobrze jest umieścić polecenie FR /SAVE w zbiorze AUTOEXEC.BAT na twardym dysku.

Podanie polecenia FR C: spowoduje usunięcie skutków formatowania (jeżeli przedtem informacje o stanie dysku zapisywane były poleceniem FR C: /SAVE) i udostępni z powrotem dane, zapisane na twardym dysku przed formatowaniem.

QU

Zbiór QU.EXE służy do ratowania skasowanych zbiorów (tzw. odkasowywanie). Ponieważ skasowanie wcale nie polega na dokładnym wymazaniu zawartości, a jedynie na zmianie statusu z "istniejący" na "skasowany", w prosty sposób można odzyskać dane zawarte w przypadkowo skasowanym zbiorze. Niestety, zmiana statusu zbioru wiąże się z utraceniem pierwszego znaku z jego

nazwy. Dlatego od użytkownika zawsze będzie wymagane uzupełnienie pierwszej litery nazwy odzyskiwanego zbioru.
Polecenie QU służy właściwie do odzyskiwania zbiorów w prostych przypadkach, gdy nie została zniszczona informacja o zbiorze (przez zbiory zapisane później, na skasowanym zbiorze) i zawarta w zbiorze. Program jest uruchamiany przez podanie polecenia w postaci

QU [*nzwzbru*[.*roz*]] [/a]

Użycie parametru /a powoduje rozpoczęcie pracy programu w trybie automatycznym (pierwsze litery wszystkich odzyskiwanych zbiorów zostaną zastąpione literą a), pominięcie tego parametru spowoduje rozpoczęcie pracy w trybie konwersacyjnym. I tak np. podanie polecenia QU \ACAD*.DWG /A spowoduje odzyskanie w katalogu \ACAD wszystkich skasowanych zbiorów, których nazwa miała rozszerzenie DWG. Nazwy wszystkich zbiorów będą rozpoczynać się od litery a. Podanie polecenia QU \ACAD*.DWG będzie miało ten sam efekt, z tym, że użytkownik zostanie poproszony o podanie pierwszej litery nazwy każdego z odzyskiwanych zbiorów.

SD

Zbiór SD.EXE służy do generowania raportów o fragmentacji zbiorów na dysku i jej usuwaniu. Polecenie SD jest podawane w postaci

SD [*d:*]

lub

SD [*obszar_działania*] /report [/s][/p][/l]

SD [*d:*] powoduje usunięcie fragmentacji zbiorów na wskazanym dysku.
Użycie parametru /report powoduje wygenerowanie raportu o stanie fragmentacji zbiorów we wskazanym *obszarze działania* polecenia SD.
Użycie parametru /s powoduje uwzględnienie w raporcie podkatalogów wskazanego *obszaru działania*.

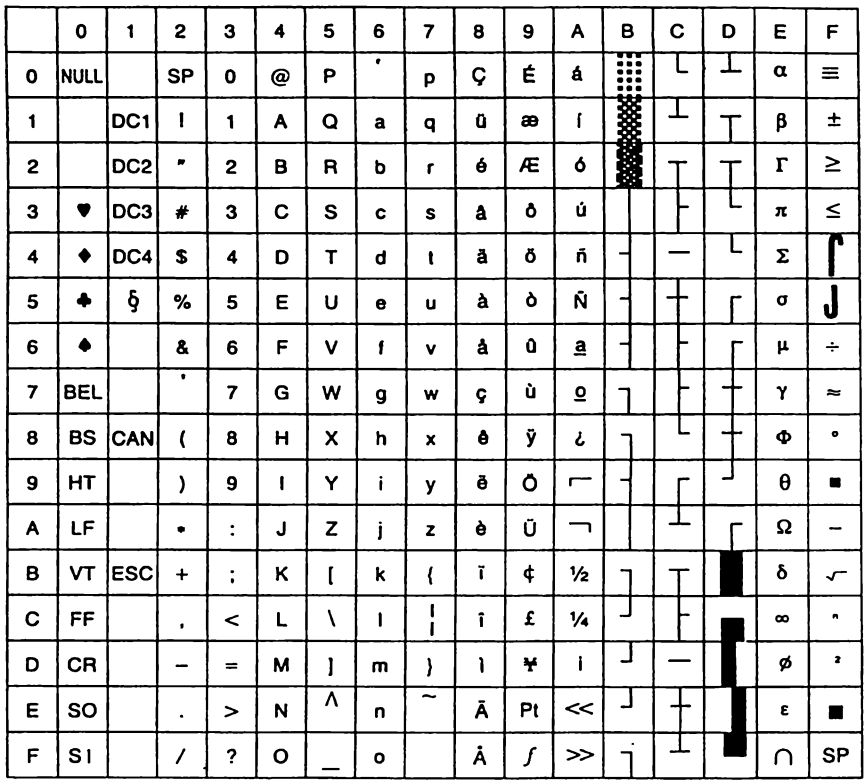
Użycie parametru /p ma taki sam skutek, jak dla polecenia **DIR**.
Użycie parametru /t powoduje wyprowadzenie procentowej informacji o ilości nieciągłych zbiorów na dysku.

Podczas działania polecenia SD [*d:*] nie wolno wyłączać komputera (zabroniony jest zimny i gorący start), polecenia tego nie można też uruchamiać ze zbioru typu BAT. SD nie może współpracować z programami rezydentnymi, np. SideKick czy PCTools. Dlatego przed uruchomieniem SD należy usunąć takie programy z pamięci. Wszystkie te ograniczenia wynikają z faktu, że SD odczytuje i zapisuje informacje na twardym dysku w sposób ciągły. Jakiekolwiek przerwanie pracy SD (np. przez program rezydentny) mogłoby zniszczyć nawet całą tę informację.

D	H	Z	D	H	Z	D	H	Z	D	H	Z	D	H	Z
0	00	NUL	52	34	4	104	68	h	156	9C		208	D0	
1	01	SOH	53	35	5	105	69	i	157	9D		209	D1	
2	02	STX	54	36	6	106	6A	j	158	9E		210	D2	
3	03	ETX	55	37	7	107	6B	k	159	9F		211	D3	
4	04	EOT	56	38	8	108	6C	l	160	A0	á	212	D4	
5	05	ENQ	57	39	9	109	6D	m	161	A1	í	213	D5	
6	06	ACK	58	3A	:	110	6E	n	162	A2	ó	214	D6	
7	07	BEL	59	3B	;	111	6F	o	163	A3	ú	215	D7	
8	08	BS	60	3C	<	112	70	p	164	A4	Ñ	216	D8	
9	09	HT	61	3D	=	113	71	q	165	A5	ñ	217	D9	
10	0A	LF	62	3E	>	114	72	r	166	A6	ª	218	DA	
11	0B	VT	63	3F	?	115	73	s	167	A7	º	219	DB	
12	0C	FF	64	40	@	116	74	t	168	A8	¿	220	DC	
13	0D	CR	65	41	A	117	75	u	169	A9	⌐	221	DD	
14	0E	SO	66	42	B	118	76	v	170	AA	¬	222	DE	
15	0F	SI	67	43	C	119	77	w	171	AB	½	223	DF	
16	10	DLE	68	44	D	120	78	x	172	AC	¼	224	E0	α
17	11	DC1	69	45	E	121	79	y	173	AD	¡	225	E1	ß
18	12	DC2	70	46	F	122	7A	z	174	AE	«	226	E2	Γ
19	13	DC3	71	47	G	123	7B	{	175	AF	»	227	E3	π
20	14	DC4	72	48	H	124	7C	¦	176	B0	▒	228	E4	Σ
21	15	NAK	73	49	I	125	7D	}	177	B1	▓	229	E5	σ
22	16	SYN	74	4A	J	126	7E	~	178	B2	│	230	E6	µ
23	17	ETB	75	4B	K	127	7F	DEL	179	B3	┤	231	E7	τ
24	18	CAN	76	4C	L	128	80		180	B4	╡	232	E8	Φ
25	19	EM	77	4D	M	129	81		181	B5	╢	233	E9	Θ
26	1A	SUB	78	4E	N	130	82		182	B6	╖	234	EA	Ω
27	1B	ESC	79	4F	O	131	83		183	B7	╕	235	EB	δ
28	1C	FS	80	50	P	132	84		184	B8	╣	236	EC	∞
29	1D	GS	81	51	Q	133	85		185	B9	║	237	ED	∅
30	1E	RS	82	52	R	134	86		186	BA	╗	238	EE	∈
31	1F	US	83	53	S	135	87	BEL	187	BB	╝	239	EF	∩
32	20	SPC	84	54	T	136	88	BS	188	BC	╜	240	F0	≡
33	21	!	85	55	U	137	89	HT	189	BD	╛	241	F1	±
34	22	"	86	56	V	138	8A	LF	190	BE	┐	242	F2	≥
35	23	#	87	57	W	139	8B	VT	191	BF	└	243	F3	≤
36	24	$	88	58	X	140	8C	FF	192	C0	┴	244	F4	⌠
37	25	%	89	59	Y	141	8D	CR	193	C1	┬	245	F5	⌡
38	26	&	90	5A	Z	142	8E	SO	194	C2	├	246	F6	÷
39	27	'	91	5B	[	143	8F	SI	195	C3	─	247	F7	≈
40	28	(	92	5C	\	144	90		196	C4	┼	248	F8	°
41	29	)	93	5D	]	145	91	DC1	197	C5	╞	249	F9	∙
42	2A	*	94	5E	^	146	92	DC2	198	C6	╟	250	FA	·
43	2B	+	95	5F	_	147	93	DC3	199	C7	╚	251	FB	√
44	2C	,	96	60	`	148	94	DC4	200	C8	╔	252	FC	n
45	2D	-	97	61	a	149	95		201	C9	╩	253	FD	2
46	2E	.	98	62	b	150	96		202	CA	╦	254	FE	■
47	2F	/	99	63	c	151	97		203	CB	╠	255	FF	
48	30	0	100	64	d	152	98	CAN	204	CC	═			
49	31	1	101	65	e	153	99	EM	205	CD	╬			
50	32	2	102	66	f	154	9A		206	CE	╧			
51	33	3	103	67	g	155	9B	ESC	207	CF	╨			

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NULL		SP	0	@	P	`	p								
1		DC1	!	1	A	Q	a	q		DC1						
2		DC2	"	2	B	R	b	r		DC2						
3		DC3	#	3	C	S	c	s		DC3						
4		DC4	$	4	D	T	d	t		DC4						
5			%	5	E	U	e	u								
6			&	6	F	V	f	v								
7	BEL		'	7	G	W	g	w	BEL							
8	BS	CAN	(	8	H	X	h	x	BS	CAN						
9	HT		)	9	I	Y	i	y	HT							
A	LF		*	:	J	Z	j	z	LF							
B	VT	ESC	+	;	K	[	k	{	VT	ESC						
C	FF		,	<	L	\	l	¦	FF							
D	CR		-	=	M	]	m	}	CR							
E	SO		.	>	N	^	n	~	SO							
F	SI		/	?	O	_	o	DEL	SI							

Zestaw znaków graficznych IBM-G1



Zestaw znaków graficznych IBM-G2

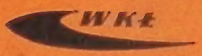
DODATEK C. SEKWENCJE ZNAKÓW STERUJĄCYCH EPSON ESC/P

Sekwencje znaków sterujących podane w poniższej tabeli są zgodne ze standardem ESC/P firmy EPSON. Firma ta już od dłuższego czasu usiłuje (z powodzeniem) narzucić swoje rozwiązania w programowym sterowaniu drukarkami i zlikwidować bałagan do tej pory panujący w tej dziedzinie. Ponieważ szczegółowy opis funkcji realizowanych na określonej drukarce przez daną sekwencję kodów sterujących na pewno jest zawarty w instrukcji do drukarki, przedstawione tutaj zestawienie ma charakter poglądowo-informacyjny i może posłużyć np. do sprawdzenia zgodności kupowanej drukarki ze standardem EPSONa.

Sekwencja	Funkcja
CR	rozpoczęcie drukowania
LF	wysunięcie papieru o jeden wiersz
FF	wysunięcie papieru do początku nast. strony
VT	tabulacja pionowa
HT	tabulacja pozioma
SO	włączenie pisma poszerzonego
SI	włączenie pisma zagęszczonego
BEL	sygnał dźwiękowy
DC2	wyłączenie pisma zagęszczonego
DC4	wyłączenie pisma poszerzonego
CAN	skasowanie danych z bufora drukarki
DEL	skasowanie ostatniego znaku z bufora drukarki
BS	cofnięcie głowicy drukarki o jeden znak
ESC space	zmiana odstępu między znakami
ESC !	ustawienie trybu drukowania
ESC %	przełączenie generatora znaków na ROM lub RAM
ESC :	skopiowanie zawartości ROMu do RAMu

ESC &	rozpoczęcie definiowania znaku
ESC *	ustawienie trybu grafiki bitowej
ESC /	wybór numeru kanału tabulacji pionowej
ESC \	ustawienie względnej pozycji głowicy
ESC -	włączenie / wyłączenie podkreślenia
ESC <	drukowanie jednokierunkowe z lewa na prawo
ESC >	ustawienie najstarszego bitu (MSB) na 1
ESC =	ustawienie najstarszego bitu (MSB) na 0
ESC #	unieważnienie uprzedniego ustawienia MSB
ESC @	inicjalizacja drukarki
ESC ?	zredefiniowanie trybu graficznego
ESC 0	ustawienie odstępu między wierszami na 1/8"
ESC 2	ustawienie odstępu między wierszami na 1/6"
ESC 3	ustawienie odstępu między wierszami na n/180"
ESC 4	włączenie kursywy
ESC 5	wyłączenie kursywy
ESC 8	wyłączenie czujnika końca papieru
ESC 9	włączenie czujnika końca papieru
ESC A	ustawienie odstępu między wierszami na n/72"
ESC B	ustawienie pozycji tabulatorów pionowych
ESC C	ustawienie długości strony w wierszach
ESC D	ustawienie pozycji tabulatorów poziomych
ESC E	włączenie pisma wyróżnionego
ESC F	wyłączenie pisma wyróżnionego
ESC G	włączenie pisma wytłuszczonego
ESC H	wyłączenie pisma wytłuszczonego
ESC J	jednokrotne wysunięcie papieru o n/180"
ESC K	włączenie trybu grafiki bitowej o pojedynczej gęstości
ESC L	włączenie trybu grafiki bitowej o podwójnej gęstości
ESC M	włączenie pisma typu 'elite'
ESC N	włączenie omijania perforacji
ESC O	wyłączenie omijania perforacji
ESC P	włączenie pisma typu 'pica'
ESC Q	ustawienie końca kolumny
ESC R	wybór narodowego zestawu znaków
ESC S	włączenie pisma podniesionego / opuszczonego
ESC T	wyłączenie pisma podniesionego / opuszczonego
ESC U	przełącznik drukowania jednokierunkowego
ESC W	włączenie / wyłączenie pisma powiększonego
ESC Y	włączenie trybu grafiki bitowej o podwójnej gęstości, drukowanej z podwójną prędkością
ESC Z	włączenie trybu grafiki o poczwórnej gęstości
ESC b	ustawienie pozycji tabulatorów pionowych w danym kanale
ESC p	włączenie / wyłączenie pisma proporcjonalnego
ESC s	włączenie / wyłączenie drukowania z połową prędkości
ESC x	włączenie / wyłączenie pisma NLQ

Cena zł 650.–



ISBN 83-206-0843-0